ZX-INTERFACE 1 Y ZX-MICRODRIVES:

QUÉ SON, PARA QUÉ SIRVEN Y CÓMO SE USAN

Vicente Mendoza



VENTAMATIC

Editado por: Ventamatic
Avda. de Rhode, n.º 253
ROSAS (Gerona)
Impresión: Gamma Graf
Ctra. Cuatro Caminos Km. 3,100, Nave 8
Sta. Coloma de Cervelló (Barcelona)
ISBN:
Depósito Legal: B-
Impreso en España - Printed in Spain

VICENTE MENDOZA LAMA

EL ZX-INTERFACE 1 Y LOS ZX-MICRODRIVES: QUE SON, PARA QUE SIRVEN Y COMO SE USAN

EDITADO POR VENTAMATIC

INDICE

PROLOGO

Con este libro, el usuario del ZX-SPECTRUM que disponga también del ZX-INTERFACE 1 y del ZX-MICRODRIVE, podrá aprender a manejarlo utilizando al máximo todas sus fantásticas posibilidades.

A través de una estructuración del libro perfectamente planteada, el usuario del ZX-SPECTRUM y los ZX-MICRODRIVES irá avanzando progresivamente, mediante numerosos ejemplos prácticos, en el conocimiento de la programación y utilización de este nuevo sistema de almacenamiento de datos en masa que es el ZX-MICRODRIVE, cuya velocidad y fiabilidad es inconmensurablemente superior a la de un cassette normal. Además, aprenderá a desarrollar y usar nuevos sistemas informáticos de comunicación con periféricos de todo tipo como impresoras, plotters, modems, etc. e incluso con otros ordenadores, incluyendo otros ZX-SPECTRUM, a través de los INTERFACES SERIE RS232 y de RED LOCAL incorporados en el ZX INTERFACE 1.

Y todo ello, incluso desde CODIGO MAQUINA.

Josep-Oriol Tomas i Huynh-Quan-Chiêu

A MARY

Por su apoyo y
paciencia

INTRODUCCION

Ante la reciente aparición en nuestro país de los esperados Microdrives de Sinclair y dada la poca claridad con que se explica su funcionamiento en el manual que los acompaña, creí oportuno crear este manual con el fin de plasmar en él la mayor información que me fuera posible y aclarar al máximo el comportamiento de este revolucionario sistema de almacenamiento.

Este manual está desarrollado sobre la base de que el lector posee unos conocimientos básicos de uso de los comandos del BASIC de su Spectrum. Es obvio que para pretender usar a fondo un periférico, primero se ha de saber usar el ordenador. Lo contrario carece de lógica.

Por este motivo recomiendo al lector que repase a fondo principalmente los comandos que se usan para tratar datos y programas con el cassette antes de entrar a fondo en la lectura de este manual.

Espero que los objetivos mencionados puedan quedar cumplidos y obtenga el máximo rendimiento de su Interface 1 y Microdrive.

VICENTE MENDOZA LAMA

TRATAMIENTO DE DATOS

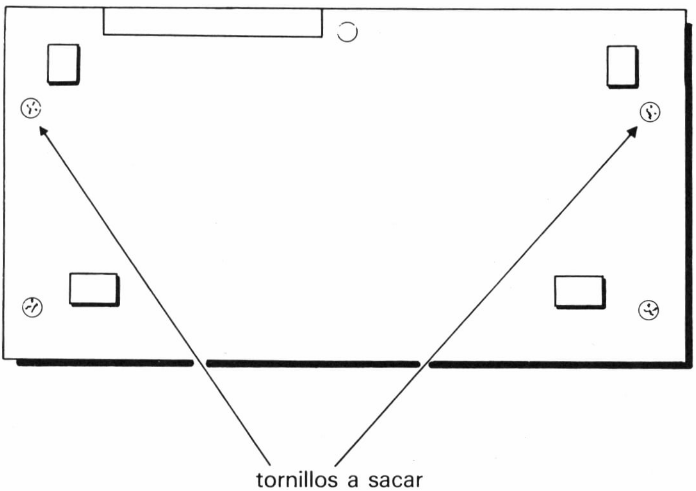
Conectando el Interface 1

Al desempaquetar su Interface 1 debe encontrar:

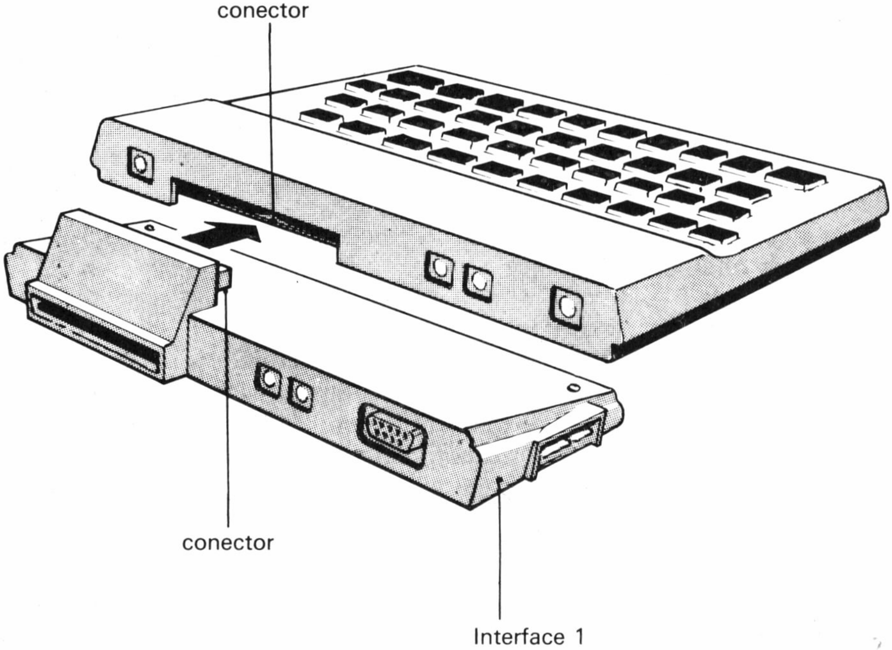
- Un manual.
- El Interface 1 (con dos tornillos).
- Un cable plano de unos 8 cm. de largo para conectar el Interface
- 1 con el **Microdrive.**
- Un cable con un jack a cada lado para usar la red local de comunicaciones (network).

Para conectar el Interface 1 con su Spectrum realice paso a paso las siguientes operaciones:

1. Desconecte la alimentación del ordenador.
2. Desconecte del **Spectrum** todos los periféricos.
3. Con un destornillador de los llamados de estrella saque los dos tornillos laterales superiores del **Spectrum**, tal como muestra la figura 1. Deje los otros tornillos como están y guarde los que ha sacado ya que sólo los necesitará si desea desconectar el Interface 1 en el futuro.



4. Conecte firmemente el Interface 1 en el conector posterior del **Spectrum** de modo que quede tal como se muestra en la figura 2.
5. Fije el Interface 1 al ordenador mediante los dos tornillos largos que venían junto al Interface 1.

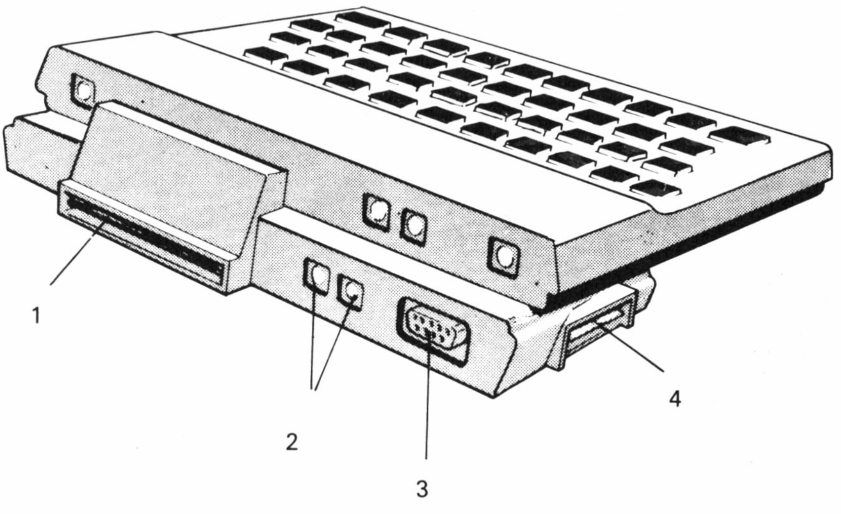


Ahora puede volver a conectar los periféricos en el conector que queda libre en el Interface 1 y conectar la alimentación al **Spectrum**.

Los conectores del Interface 1

El Interface 1 tiene cuatro tipos de conectores diferentes:

- La extensión del conector del **Spectrum**. (1)
- Los dos conectores hembra para la red local. (2)
- El conector para el RS-232 (3)
- El conector para el cable de los Microdrives. (4)

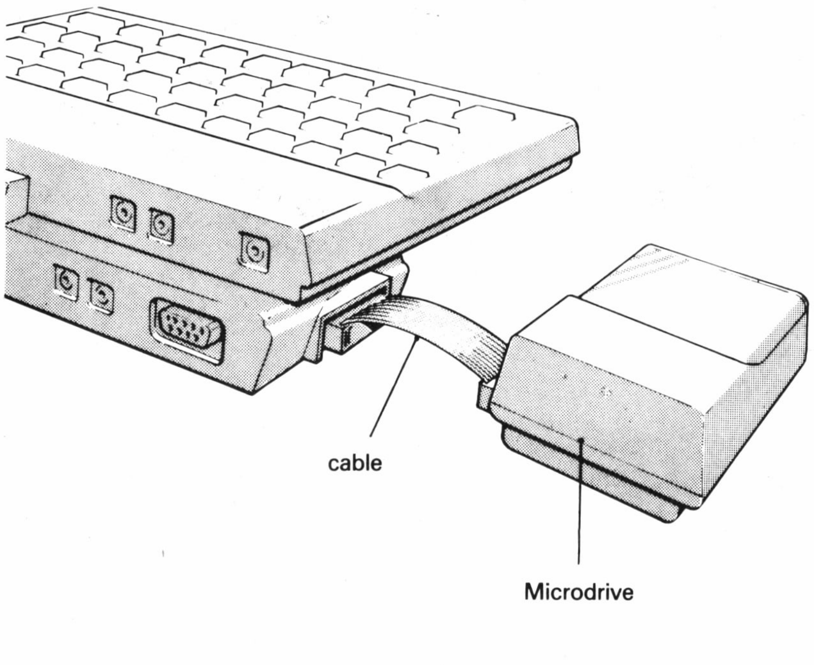


Conectando el Microdrive

Al desempaquetar el Microdrive debe encontrar:

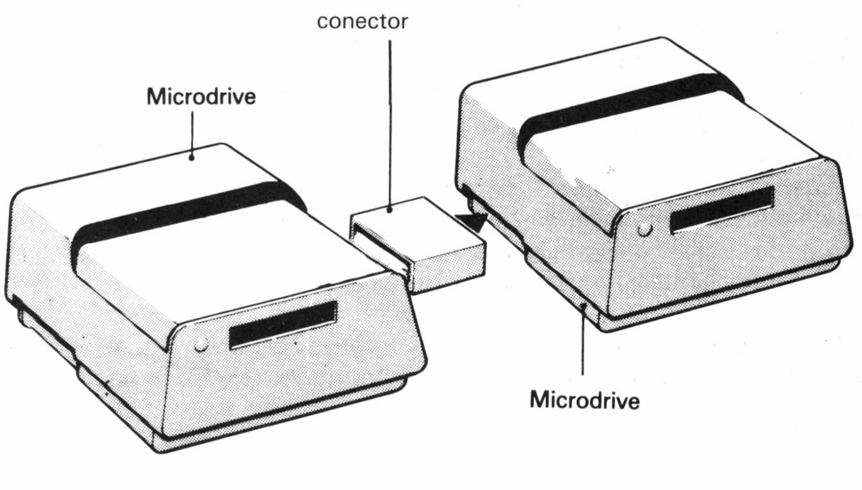
- El Microdrive.
- Un conector eléctrico de plástico rígido.
- Un cartucho de demostración.

Para conectar el Microdrive al Interface 1 inserte uno de los extremos del cable plano que venía con el Interface 1 en el conector lateral izquierdo de éste y el otro extremo en el conector lateral derecho del **Microdrive**. Debe quedar como muestra la figura 3.



Si desea conectar más Microdrives, éstos deben ser conectados entre sí como sigue:

1. Desconecte el Microdrive del Interface 1.
2. Ponga boca abajo los Microdrives y observe que cada uno lleva atornillada una placa metálica.
3. Desatornille estas placas y sáquelas de los Microdrives.
4. Coloque los Microdrives boca arriba.
5. Conecte los Microdrives entre sí por medio del conector eléctrico tal como muestra la figura 4.
6. Con cuidado vuelva a colocar los Microdrives boca abajo y coloque una de las placas metálicas de modo que una a ambos Microdrives, es decir, con dos tornillos en cada uno de ellos.
7. Coloque los Microdrives boca arriba y conéctelos de nuevo con el Interface 1.



A partir de este momento el equipo está listo para funcionar numerándose los Microdrives del 1 al 8, contando como número 1 el más cercano al Interface 1, número 2 el siguiente, etc.

Como habrá observado el Microdrive posee en la parte frontal una abertura donde se insertarán los cartuchos y un LED (lamparita roja) que se encenderá cuando la unidad esté en marcha.

Los cartuchos

Los cartuchos vienen en una cajita protectora y deben guardarse en ella cuando no se estén usando.

Cuando saque un cartucho de su funda cuide de no tocar la cinta pues podría estropearla. Observe que tiene una marca en una de sus caras, donde podrá colocar una etiqueta para identificar el cartucho.

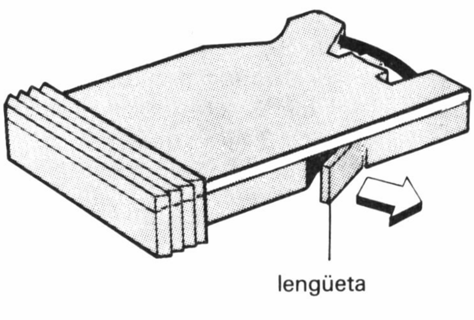
Insértelo firmemente en la ranura del Microdrive de modo que la cara con la etiqueta quede en la parte superior (a la vista mientras lo introduce). Una pequeña parte del cartucho (1/2 cm.) quedará fuera del Microdrive para posibilitar su extracción. Ahora el Microdrive y el cartucho están listos para su uso.

Cuando acabe de usarlo sáquelo con cuidado del Microdrive e introdúzcalo en su funda.

No es recomendable sacar el cartucho cuando el LED esté encendido ni conectar o desconectar la alimentación con un cartucho dentro del Microdrive.

Si desea proteger un cartucho para que no pueda borrarse lo que contiene, sencillamente rompa la lengüeta del lateral derecho del cartucho. Para poder volver a escribir en éste, coloque un trozo de celo en el lugar de la lengüeta al igual que seguramente hará con los cassettes.

Al igual que con los disquettes convencionales es muy aconsejable el tener copias de seguridad de los programas o datos y mantener los cartuchos alejados de las fuentes de magnetismo como la pantalla del TV o los altavoces de los cassettes.



Nuevas palabras

Durante la lectura de este manual podrá comprobar que aparecen algunas palabras cuyo significado puede no quedarle demasiado claro. La razón principal puede ser que son palabras del idioma inglés o que su traducción al español no define correctamente su significado.

Estas palabras son:

ARCHIVO
En general, se habla de archivo cuando se está refiriendo a un conjunto de datos.

BUFFER
Es una zona de la memoria del ordenador que se reserva para que contenga una serie de datos.

CANAL
Se habla de canal (CHANNEL) refiriéndose a un periférico o interface.

PROGRAMA
Se trata de un conjunto de instrucciones comprensibles para el ordenador.

VIA (STREAM)
Se trata del camino que ha de abrirse para que pueda circular la información a través de él hacia un canal.

LOS COMANDOS

Los nuevos comandos Basic

En el Interface 1 va alojada una nueva ROM que incorpora al **Spectrum** unos nuevos comandos para facilitar el uso del Microdrive y los nuevos canales del Interface 1.

Estos nuevos comandos son:

RUN
FORMAT
CAT
SAVE *, LOAD *, VERIFY * Y MERGE *
ERASE
MOVE
OPEN
INKEY$
INPUT
PRINT
CLS Y CLEAR

Aunque algunos de ellos ya le serán familiares, ahora se introducen nuevas opciones de uso, por lo que los veremos como si de nuevos comandos se tratara.

Antes de pasar a un estudio detallado de estos comandos puede observar cómo se comporta su Microdrive y probar su funcionamiento al mismo tiempo introduciendo el cartucho de demostración. Una vez introducido, pulse NEW en el ordenador y después teclee RUN y ENTER. Observará que el Microdrive se pone en funcionamiento y cargará los programas de demostración. Observe que en el borde de la pantalla aparecen intermitentes unas líneas negras. Esto indica que se está cargando un trozo de programa. Más adelante trataremos a fondo el modo en que están grabados los programas o los datos en el cartucho.

RUN

Este comando adquiere una nueva dimensión al ser conectado el Interface 1 al **Spectrum.**

Si se ejecuta directamente después de haber conectado la alimentación al ordenador, después de un NEW o un RANDOMIZE USR Ø, se comportará del siguiente modo:

Buscará en el Microdrive 1 un programa que se llame «run», tecleando letra a letra y en minúsculas.

Si lo encuentra lo ejecutará automáticamente, tal y como habrá podido comprobar con la cinta de demostración.

Una vez esté familiarizado con su Microdrive encontrará interesante el siguiente programa almacenado en cada cartucho para facilitarle la carga de cualquier programa.

Recuerde que este comando sólo actúa con el cartucho que esté introducido en el Microdrive 1.

```
 1Ø  REM CARGA DE PROGRAMAS CON MICRODRIVE
 2Ø  CLS : INPUT «Unidad número»; n
 3Ø  IF n < 1 OR n > 8 THEN GOTO 2Ø
 4Ø  CAT n
 5Ø  PRINT AT 5,1Ø; «PULSE PARA CARGAR»
 6Ø  PRINT AT 7,1Ø; «1 - Programa»
 7Ø  PRINT AT 9,1Ø; «2 - Bytes»
 8Ø  PRINT AT 11,1Ø; «3 - Screen$»
 9Ø  PRINT AT 13,1Ø; «4 - Matriz»
1ØØ  INPUT «Cuál selecciona?»; a
11Ø  IF a < 1 OR a > 4 THEN GOTO 1ØØ
12Ø  INPUT «Nombre»; a$
13Ø  IF LEN a$ < 1 OR a$ > 1Ø THEN GOTO 12Ø
14Ø  GOTO (a*1Ø)+14Ø
15Ø  LOAD*«m»;n;a$: STOP
16Ø  LOAD*«m»;n;a$ CODE: STOP
17Ø  LOAD*«m»;n;a$ SCREEN$: STOP
```

```
180 LOAD*«m»;n;a$ DATA a(): STOP: REM Cambie a() por la
    matriz que desee cargar.
190 REM Salvar el programa
200 SAVE*«m»;1;«run» LINE 10: VERIFY *«m»;1;«run»
210 STOP
```

Para salvarlo teclee GOTO 200 y luego compruebe su buen funcionamiento.

FORMAT

Antes de poder usar un cartucho en blanco o si desea borrarlo y poder usarlo, deberá usar este comando de la siguiente forma:

FORMAT «m»;1;«nom»

Donde:

«m» Le indica al **Spectrum** que se trata de una operación con Microdrives.

1 Le indica al ordenador que actúe sobre la unidad de Microdrives número 1

«nom» Será el nombre del cartucho. Puede ser cualquier sucesión de caracteres si su longitud no es mayor de 10.

El ordenador tarda alrededor de 45 segundos en formatear un cartucho. Durante este tiempo el borde de la pantalla presentará un flash dos veces durante esta operación y luego presentará el conocido mensaje OK.

El formateado no debe ser repetido en el mismo cartucho nunca más, a menos que se desee eliminar toda la información que contiene.

Durante esta ejecución el ordenador realiza tres operaciones:

- Borra toda la información del cartucho.
- Sectoriza el cartucho y verifica cada sector.
- Escribe el nombre del cartucho en cada sector de éste.

El cartucho se divide en sectores de 540 bytes, colocando una cabecera en cada sector con el nombre del cartucho y un bloque para los datos.

Compruebe que la operación se ha realizado correctamente tecleando CAT 1, que le devolverá el nombre del cartucho y el número de Kbytes libres en éste. Si prueba a realizar la operación FORMAT varias veces observará que el resultado en Kbytes libres es diferente cada vez que lo ejecuta por lo que es conveniente realizar esta operación varias veces hasta que se consigue un número de Kbytes libres idóneo.

La capacidad mínima garantizada por Sinclair es de 85K. Si después de varios formateos no consigue este mínimo, su cartucho es defectuoso y, según dice la garantía, le será canjeado.

A pesar de ello, normalmente debe conseguir alrededor de 85 K a 90 K libres.

CAT

Este comando le devuelve el catálogo completo del contenido de un Microdrive con la siguiente información:

- El nombre del cartucho.
- Una lista de 50 programas o filas de datos no protegidos, que contiene el cartucho.
- El total de memoria que queda libre en el Microdrive, expresado en Kbytes.

La sintaxis de esta instrucción es la siguiente:

CAT n

Donde n es el número del Microdrive a catalogar.

Además, puede enviar la salida del catálogo a una vía (stream), de la siguiente forma:

CAT m,n

Donde m es el número de la vía (stream) y n es el número de Microdrive. Esto le permite enviar el catálogo a impresora o a un archivo de datos para ser usado de algún modo.

La forma de proteger un archivo para que su nombre no salga en pantalla al usar el comando CAT, es introducir en su nombre como primer carácter CHR$ Ø como en el siguiente ejemplo:

```
10 OPEN # 5, «m»;1;CHR$ Ø +«Nombre»
20 PRINT # 5; 2
30 CLOSE # 5
```

Si prueba a ejecutar el comando CAT observará que el archivo Nombre (que es el que hemos creado), no aparece por ninguna parte a pesar de que esté en el Microdrive. Para comprobar que está, teclee:

```
10 OPEN # 5,«m»;1;«Nombre»
20 INPUT # 5,n
30 PRINT n
40 CLOSE # 5
```

En pantalla debe aparecer el número 2, que es el que almacenamos anteriormente en este archivo.

LOAD

A pesar de que este comando ya le debe resultar familiar, su sintaxis es diferente con los Microdrives. Para cargar programas, este comando se debe introducir de esta forma:

LOAD *«m»;1;«Nom»

Donde:

«m»;* Le indica al ordenador que se quiere cargar del Microdrive.
1; Indica qué unidad de Microdrive se está usando.
«Nom» Indica el nombre del programa a cargar.

Después de una breve pausa el programa debe estar cargado.

Si se trata de cargar matrices o Bytes a continuación del comando arriba mencionado deben constar las mismas expresiones que para la carga desde el cassette, es decir:

CODE Para cargar Bytes.
DATA Para cargar matrices de datos.
SCREEN$ Para cargar pantallas.

SAVE

Este comando también cambia un poco cuando ha de usarse con el Microdrive. La forma de hacerlo es:

SAVE * «M»;1; «Nom»

Donde:

«m»; Le indica al **Spectrum** que se quiere salvar en el Microdrive.
1; Indica qué unidad de Microdrive se está usando.
«Nom» Indica el nombre del programa a salvar.

Después de un breve espacio de tiempo debe aparecer en pantalla el conocido mensaje OK.

Si se trata de salvar matrices, bytes, pantallas o programas con autoejecución en una línea determinada, a continuación del comando pueden colocarse las expresiones adecuadas al igual que se hacía con el cassette, es decir:

CODE Para salvar bytes.
DATA Para salvar matrices de datos.
SCREEN$ Para salvar pantallas
LINE N Para indicar la linea (n) donde ha de empezar la ejecución del programa.

Durante la operación de salvar un programa o datos, el Microdrive realiza las siguientes maniobras:

Primero da una vuelta completa a la cinta del cartucho para comprobar que existe espacio suficiente en él para grabar el programa y que el nombre introducido es correcto y no existe otro igual en el cartucho. Después va buscando sectores libres y va grabando en cada uno de ellos 540 bytes del programa hasta completar la grabación.

Para grabar un archivo de datos y que su nombre no aparezca cuando se pide un catálogo (comando CAT), se debe incluir en su nombre el carácter con el código Ø (CHR$ Ø). Por ejemplo:

LET G$ = CHR$Ø + «Nombre»

y luego:

SAVE * «M»;1;G$

VERIFY

Al ejecutar este comando el ordenador comprueba que la grabación que acabamos de hacer con el nombre «Nom» corresponde a lo que hay en la memoria del **Spectrum**, al igual que cuando se realiza con el cassette.

Se ejecuta de este modo:

VERIFY *«m»;1; «Nom»

Donde cada una de las diferentes expresiones tiene el mismo significado que para las operaciones LOAD y SAVE y las indicaciones en el caso de bytes o pantallas son las mismas que en los casos mencionados.

Su comportamiento es similar a la operación de verificar con el cassette.

MERGE

Al ejecutar este comando el programa «Nom» se mezclará con el que esté en memoria al igual que cuando se operaba con el cassette.

Se debe ejecutar de este modo:

MERGE *«num»;1;«Nom»

Donde todas las especificaciones son las mismas que para LOAD y SAVE, por lo que no las repetiré.

ERASE

Este comando se utiliza para borrar cualquier programa o archivo de datos del Microdrive. Se utiliza según estas reglas:

ERASE «m»;1;«Nom»

Donde:

«m»;1; Indica la unidad de Microdrive a utilizar.
«Nom» Indica el nombre del programa a borrar.

Observe que en este comando no se ha de indicar al **Spectrum** que utilice los Microdrives mediante el asterisco (*) como se hacía con LOAD, SAVE, VERIFY y MERGE. Esto es debido a que esta instrucción es exclusiva para Microdrives por lo que el ordenador no duda en saber cuál es el medio de almacenamiento en el cual se va a operar.

Si durante la operación de borrado de un programa o archivo de datos para la ejecución mediante BREAK o por otra razón el programa no estará totalmente borrado, pero sí inutilizado por lo que deberá ejecutar de nuevo el comando ERASE para completar el borrado. En esta segunda ocasión el Microdrive tardará un poco más en borrarlo.

Resulta curioso observar el tiempo que tarda al intentar realizar el borrado de un programa que no existe. Ello da una idea de las operaciones que realiza para asegurarse de que no se está equivocando.

EJEMPLOS

Practicando con programas

Coloque un cartucho virgen en el Microdrive 1 y formatéelo:

FORMAT «m»;1;«Prueba 1»

Una vez aparezca el mensaje OK, pida un catálogo:

CAT 1

El cual le devolverá el nombre del cartucho y los Kbytes libres que tiene el cartucho.

Introduzca el programa que comentamos junto al comando RUN y sálvelo tal como se explica en dicho capítulo.

Luego pulse NEW y ENTER y teclee este programa:

```
10 FOR N=1 TO 100
20 PRINT N; «—»;
30 NEXT N
```

Este programa imprime en pantalla los números del 1 al 100.

Sálvelo mediante:

SAVE *«m»;1;«Numeros»
Verifique esta operación:
VERIFY *«m»; 1; «Numeros»

Si el ordenador devuelve el mensaje OK, pulse NEW y ENTER y teclee este otro programa:

```
100 FOR N=1 TO 100
110 PRINT N*N;«—»;
120 NEXT N
```

El cual imprime los cuadrados de los números del 1 al 100.

Sálvelo:

SAVE *«m»;1;«Cuadrados»

Verifique:

VERIFY *«m»;1; «Cuadrados»

Pulse ahora NEW y ENTER y después:

LOAD *«m»;1; «Números»

Y se cargará el primero de los programas que hemos grabado. Teclee:

MERGE *«m»;1; «Cuadrados»

y luego LIST y ENTER para comprobar que tiene en memoria los dos programas que hemos grabado. Pulse:

SAVE *«m»;1; «Los dos»

y después pida un catálogo con CAT 1. Tendrá en el Microdrive los tres programas: «Números»; «Cuadrados» y «Los dos». Borremos los dos primeros:

ERASE «m»;1; «Numeros»
ERASE «m»;1; «Cuadrados»

Compruébelo mediante CAT 1. Pulse NEW y RUN, para cargar el programa del capítulo en el que tratamos el comando RUN. Introdúzcale la unidad 1. Luego pulse el 1 para cargar programas e introduzca el nombre «Los dos».

Con esto habremos comprobado todos los comandos que hemos visto hasta ahora. Realice Vd. todas las pruebas que crea necesarias hasta tener un buen dominio del funcionamiento del Microdrive con programas.

Si tiene alguna duda con el significado y el resultado de alguna de estas expresiones en su concepto básico, es decir, con su uso a través de cassette, consulte el manual que se le entregó con el **Spectrum**, ya que es muy importante tener un buen dominio de estos comandos antes de profundizar en su uso con los Microdrives.

AGENDA

AGENDA

Se incluye este programa como ejemplo de uso de los Microdrives con el manejo de bases de datos.

Puesto que el programa es meramente orientativo, no está aprovechada toda la memoria y permite ser modificado fácilmente para adecuarlo a las necesidades del usuario.

El programa consiste en el mantenimiento de una agenda con diferentes fichas que pueden contener nombre, dirección, población, provincia y teléfono en cada uno de los registros.

Para ello se define la ficha en la línea 9900 en forma de sentencias DATA según las siguientes normas:

1. Primero debe colocar el nombre de cada campo, o parte de la ficha (Nombre, etc.)
2. Luego debe colocar la longitud, o sea, el número de caracteres que puede ocupar ese campo.
3. Seguidamente debe indicar la línea donde debe aparecer en pantalla cuando se quiera consultar, dar de alta, etc.
4. A continuación debe indicar el carácter a partir del cual comienza el campo, que debe coincidir con la suma de la longitud del campo y el carácter de comienzo del campo anterior.

Por ejemplo, si desea que el campo «Nombre» tenga una longitud de 20 caracteres y se presente en pantalla en la línea 5, la línea 9900 debe empezar:

9900 DATA «Nombre», 20, 5, 1,......

Por supuesto, debe tener en cuenta la longitud total del registro para poder dimensionar la matriz de acuerdo con ella. El dimensionado de la matriz que contiene los datos se realiza en la línea 30, después de asignar a la variable «tot» el número de fichas que quepan en memoria con la longitud deseada, teniendo en cuenta que el programa usa la última ficha y debe permanecer vacía. En el ejemplo de la Agenda se dimensiona una matriz que le permite tener hasta 180 fichas en memoria de 195 caracteres cada una, aunque si elimina las líneas donde se reali-

zan comentarios (REM) y algunas otras donde se realizan comprobaciones a los INPUTs, podrá conseguir un número mayor de fichas.

La estructura del programa es la siguiente:

Líneas 10 a 140

Inicializa las variables, presenta en pantalla el menú y espera a que se pulse la opción deseada, en función de la cual va a una de las diferentes subrutinas que contiene el programa. Al volver de esta, vuelve a presentar el menú y se repite el proceso.

Líneas 1000 a 1300

En esta subrutina se realiza la entrada de los datos de una nueva ficha. En principio se le pide al usuario que entre el número de ficha después de informarle de los límites a los cuales ha de acogerse y se chequea si el número de ficha es correcto o es 0. Si es 0 se vuelve al menú.

A partir de la línea 1100 se realiza la lectura de la ficha y se envía el programa a la subrutina de entrada de datos de la línea 9200. Una vez entrada toda la ficha se envía el programa a la subrutina de la línea 9300 donde se pide la conformidad de los datos la cual queda reflejada en la variable OK. Si ésta es 1 se da la ficha como válida y se vuelve a la línea 1000. En caso contrario se envía el programa a la subrutina que muestra la ficha y luego a la subrutina que corrige un campo en concreto. Después se vuelve al chequeo de la variable OK, comenzando el proceso de nuevo.

Líneas 2000 a 2080

En esta subrutina se permite al usuario realizar cualquier modificación a una ficha. Para ello se le pide primero el número de la ficha y se chequea para ver si es correcto. Luego se envía el programa a la subrutina de mostrar la ficha y a la de corregir cualquier campo y se pide la conformidad de los datos igual que en la entrada de estos. En caso de que sean conformes se vuelve a la línea 2000 y en caso contrario vuelve a comenzar el proceso.

Líneas 3000 a 3170

En esta subrutina se permite realizar la consulta de una ficha en concreto sin modificarla. Para ello se pide al usuario que entre el número

de ficha y se envía el programa a la subrutina que muestra la ficha tras lo cual se vuelve al principio de esta subrutina.

Líneas 4000 a 4100
Esta subrutina permite el borrado de una ficha. Después de haber introducido el número de la ficha y de comprobar que sea correcto se muestra la ficha en pantalla y se pregunta si es la que se desea borrar. En caso afirmativo se borra y se vuelve a la línea 4010. En caso de error se vuelve a la línea 4000 sin haber borrado la ficha.

Líneas 5000 a 5160
En esta subrutina se realiza una selección de entre el total de las fichas si el campo que se indica es igual a lo que el usuario ha entrado. Ello permite, por ejemplo, seleccionar todas las fichas de una misma población, etc.

Líneas 6000 a 6060
Esta subrutina se encarga de cargar del Microdrive un archivo completo. Para ello pide el nombre del archivo y lo busca en la unidad de Microdrive número 1.

Líneas 7000 a 7070
Esta subrutina permite grabar el fichero completo bajo cualquier nombre. Antes de realizar la grabación, el programa borra del cartucho cualquier fichero o programa con el nombre que le ha introducido. Si no existe ninguno realizará la operación correctamente. Esto es una medida de precaución para evitar errores dentro del programa.

Líneas 8000 a 8050
Esta rutina es la de fin de programa y le recuerda que ha de grabar el fichero si no lo ha hecho. En caso de que desee hacerlo, vuelve a la línea 7000. En caso contrario el programa finaliza.

Líneas 9000 a 9360
En estas líneas se encuentran las subrutinas de impresión en pantalla de la ficha, la de entrada de datos y la de corrección.

```
  10 REM   **  AGENDA  **
  11 REM
  12 REM
  13 REM   § VICENTE MENDOZA
  14 REM
  15 REM
  16 REM
  17 REM VERSION PARA 48 K
  18 REM
  19 REM
  20 LET v$="___________________
______________": LET u$="
                                "
  30 LET d$="": LET t$="": LET t
ot=180: DIM b$(tot+1,195)
  50 REM Menu
  60 CLS #: BORDER 1: PRINT AT 1
,12; PAPER 1; INK 7;" MENU "
  70 PRINT AT 4,3; BRIGHT 1;"1 -
 Entrar nueva ficha.";AT 6,3;"2
- Modificar una ficha.";AT 8,3;"
3 - Consultar una ficha.";AT 10,
3;"4 - Borrar una ficha.";AT 12,
3;"5 - Seleccionar fichas."
  80 PRINT AT 14,3; BRIGHT 1;"6
- LOAD fichas.";AT 16,3;"7 - SAV
E fichas.";AT 18,3;"8 - Fin."
  90 PRINT AT 21,10;"OPCION: ";
FLASH 1;"?"
 100 IF INKEY$="" THEN  GO TO 10
0
 110 LET w$=INKEY$: IF w$<"1" OR
 w$>"8" THEN  GO TO 100
 120 PRINT AT 21,18; INVERSE 1;C
ODE w$-48
 130 BEEP .5,10: BEEP .5,20: BEE
P .5,30
 140 GO SUB 1000*(VAL w$)
 150 GO TO 50
1000 REM Entrada de una ficha
1010 CLS : PRINT AT 1,7; INVERSE
```

```
 1;"CREACION DE FICHAS": PRINT A
T 7,5;" Las fichas van numeradas
   de 1 a 180."''"      Por favor
 entre el numero   de la ficha a
 crear."
1020 PRINT '"      Entre el numer
o ""0"" para    volver al MENU."
1030 INPUT "Ficha numero ";f
1040 IF f=0 THEN  RETURN
1050 IF f<1 OR f>tot THEN  GO TO
 1030
1060 FOR o=0 TO 50: NEXT o
1080 CLS
1090 PRINT AT 1,10; INVERSE 1;"
FICHA: ";f
1100 RESTORE 9900: FOR n=1 TO 6:
 READ q$,l,c,m: LET t$=""
1110 GO SUB 9200
1120 NEXT n
1230 GO SUB 9300
1240 IF ok=1 THEN  GO TO 1000
1250 GO SUB 9100: GO SUB 9300: G
O TO 1240
1300 GO TO 1000
2000 REM Modificaciones
2010 CLS : PRINT AT 1,10; INVERS
E 1;"MODIFICAS"
2020 INPUT "Num. ficha (O-MENU)
? ";f
2030 IF f=0 THEN  RETURN
2040 IF f<1 OR f>tot THEN  GO TO
 2020
2050 PRINT AT 21,0;"MODIFICAS FI
CHA NUM. ";f
2060 GO SUB 9100: GO SUB 9300
2070 IF ok=1 THEN  GO TO 2000
2080 GO TO 2060
3000 REM Consultas
3110 CLS : PRINT AT 1,10; INVERS
E 1;"CONSULTAS"
3120 INPUT "Num. ficha (O-MENU)
```

```
? ";f
3130 IF f=0 THEN  RETURN
3140 IF f<1 OR f>tot THEN  GO TO
 3120
3150 PRINT AT 21,0;"CONSULTAS FI
CHA NUM. ";f
3160 GO SUB 9100
3170 GO TO 3120
4000 REM Borrado de una ficha
4005 CLS
4010 PRINT AT 1,13; INVERSE 1;"B
ORRAS"
4020 INPUT "Num. ficha (0-MENU)
? ";f
4030 IF f=0 THEN  RETURN
4040 IF f<1 OR f>tot THEN  GO TO
 3120
4050 PRINT AT 21,0;"BORRAS FICHA
 NUM. ";f;"          "
4060 GO SUB 9100
4070 INPUT "Estas seguro ? "; LI
NE w$: IF w$="" THEN  GO TO 4070
4080 IF w$( TO 1)<>"s" AND w$( T
O 1)<>"S" THEN  GO TO 4000
4090 LET b$(f)="": PRINT AT 21,0
;"FICHA NUM. ";f;" "; FLASH 1;"
BORRADA "
4100 GO TO 4010
5000 REM Seleccion por campos
5010 CLS
5020 PRINT AT 1,10; INVERSE 1;"S
ELECCION"
5030 LET f=tot+1: GO SUB 9100
5040 INPUT "Num. del campo a sel
eccionar ?  (0=MENU)";campo: IF
campo<0 OR campo>6 THEN  GO TO 5
040
5045 IF campo=0 THEN  RETURN
5050 INPUT "Igual a ... (tal com
o lo entro) ? ";w$
5060 RESTORE 9900: FOR n=1 TO ca
```

```
mpo: READ q$,l,c,m: NEXT n
5070 CLS
5080 PRINT AT 21,0;"CAMPO ";camp
o;"= ";w$
5090 PRINT AT 1,10; INVERSE 1;"
BUSCANDO ";AT 3,0;"En fichas num
s. :": PRINT
5100 REM Selecciona
5110 FOR n=1 TO tot
5120 IF b$(n,m TO m+l-1)=w$+u$(
TO l-LEN w$) THEN  PRINT " "; IN
VERSE 1;n; INVERSE 0;" ";
5130 NEXT n
5140 PRINT AT 21,20; FLASH 1;" F
IN ": BEEP 1,0
5150 PRINT AT 21,0;"Pulsa una te
cla ": PAUSE 0
5160 GO TO 5000
6000 REM Load de fichas del
         Microdrive
6010 CLS : PRINT AT 1,7; INVERSE
 1;"CARGA DE FICHAS": CAT 1
6020 INPUT "Nombre del fichero";
w$: IF w$="" OR LEN w$>10 THEN
GO TO 6020
6030 PRINT AT 21,10; FLASH 1;"CA
RGANDO"
6040 LOAD *"m";1;w$ DATA b$()
6050 PRINT AT 21,10; FLASH 1;"CO
RRECTO": BEEP 2,10
6060 RETURN
7000 REM Save de fichas al
         Microdrive
7010 CLS : PRINT AT 1,7; INVERSE
 1;"SAVE DE FICHAS": CAT 1
7020 INPUT "Nombre del fichero";
w$: IF w$="" OR LEN w$>10 THEN
GO TO 6020
7030 PRINT AT 21,3;"BORRANDO FIC
HERO ANTERIOR": ERASE "m";1;w$
7040 PRINT AT 21,3;"SALVANDO NUE
```

```
VO FICHERO    ": SAVE *"m";1;w$ D
ATA b$()
7050 PRINT AT 21,3;"VERIFICANDO
NUEVO FICHERO": VERIFY *"m";1;w$
 DATA b$()
7060 PRINT AT 21,0; FLASH 1;"
    OPERACION CORRECTA        ":
BEEP 2,0
7070 RETURN
8000 REM Fin
8010 CLS : PRINT AT 10,5;"Recuer
de que ha de grabar  su fichero
antes de finalizar   si no quier
e perder todos los   datos."
8020 INPUT "Desea grabar el fich
ero  s/n ?  ";w$: IF w$="" THEN
 GO TO 8020
8030 IF w$( TO 1)="s" OR w$( TO
1)="S" THEN  GO SUB 7000
8040 CLS #: PRINT AT 10,12; FLAS
H 1;" F I N "
8050 STOP
9000 REM Subrutinas
9005 REM Espera una tecla en la
         entrada de datos
9010 IF INKEY$="" THEN  GO TO 90
10
9020 LET d$=INKEY$: IF CODE d$=1
3 THEN  PRINT AT C+2,(LEN T$);"
  ": LET t$=t$+u$( TO (1-LEN t$)
): BEEP .5,10: GO TO 9040
9021 IF CODE d$=8 THEN  IF LEN t
$>=1 THEN  LET z$=t$: LET t$=z$(
 TO (LEN t$-1)): PRINT AT c+1,1;
t$;"_";AT c+2,(LEN t$)+1;"^    ":
 FOR O=0 TO 20: NEXT O: GO TO 90
00
9022 IF CODE d$=8 AND LEN t$<1 T
HEN  GO TO 9000
9023 IF CODE d$<32 OR CODE d$>16
4 THEN  GO TO 9000
```

```
9025 LET t$=t$+d$: PRINT AT c+2,
(LEN t$);" ^ "
9030 IF INKEY$<>"" THEN  GO TO 9
030
9040 RETURN
9100 REM Mostrar la ficha
9110 RESTORE 9900: FOR n=1 TO 6:
 READ q$,l,c,m
9120 PRINT AT c,1; PAPER 2; INK
6;n; PAPER 7; INK 0; INVERSE 1;q
$;AT c+1,1; INVERSE 0;b$(f,m TO
m+l-1)
9130 NEXT n
9140 RETURN
9200 REM Colocacion en pantalla
9210 PRINT AT c,1; PAPER 2; INK
6;n; PAPER 7; INK 0; INVERSE 1;q
$;AT c+1,1; INVERSE 0;v$( TO l);
AT c+2,1;"^"
9220 GO SUB 9000: PRINT AT c+1,1
;t$
9230 IF LEN t$<l THEN  GO TO 922
0
9240 PRINT AT c+2,(LEN t$);"  "
9250 LET b$(f,m TO m+l-1)=t$: LE
T t$="": LET d$=""
9270 RETURN
9300 INPUT "Datos correctos s/n
"; LINE w$: IF w$="" OR CODE w$=
13 THEN  GO TO 9300
9310 IF w$( TO 1)="s" OR w$( TO
1)="S" THEN  LET ok=1: RETURN
9320 IF w$( TO 1)<>"n" AND w$( T
O 1)<>"N" THEN  GO TO 9300
9330 INPUT "Numero del campo err
oneo ? "; LINE w$
9334 IF w$="" THEN  GO TO 9330
9335 LET w=VAL w$: IF w<1 OR w>6
 THEN  GO TO 9330
9340 RESTORE 9900: FOR n=1 TO VA
L w$: READ q$,l,c,m: NEXT n: CLS
```

```
 : LET n=n-1: PRINT AT 21,7; FLA
SH 1;"CORRIGIENDO DATOS": GO SUB
 9200
9350 CLS : GO SUB 9100: GO TO 93
00
9360 RETURN
9899 REM Ficha
9900 DATA "Nombre",20,5,1,"Apell
idos",30,7,21,"Direccion",30,9,5
1,"Poblacion",30,11,81,"Provinci
a",30,13,111,"Telefono",15,15,14
1
```

TRATAMIENTO DE DATOS

Datos, Canales y Streams

Como datos se entienden una serie de letras, números o símbolos, que son operados o utilizados por un programa.

Los datos pueden ser enviados o recibidos desde y hacia varias partes de las que componen su equipo (ordenador y periféricos). Cada una de estas partes suele ser llamada CANAL. Así pues los canales hacia donde la unidad central puede enviar datos pueden ser todos o algunos de los siguientes:

- a la pantalla.
- a la Impresora ZX.
- a un archivo en el Microdrive, con objeto de almacenarlos.
- a otro ordenador (**Spectrum**), por medio de la red local.
- a cualquier periférico (modem, impresora, etc.), por medio del Interface RS-232.

Y los canales desde donde puede recibir datos pueden ser:

- el teclado.
- un archivo en el Microdrive.
- otro **Spectrum**, por medio de la red local.
- por el Interface RS-232, de un modem o terminal.

Los caminos que siguen estos datos tanto en un sentido como en otro deben ser abiertos por medio de las vías (streams). En el **Spectrum** dispone de 16 vías (streams) numeradas del 0 al 15 y el número de vía (stream) debe ser precedido siempre por el símbolo #.

Por si no le ha quedado claro le pondré un ejemplo. Suponga que Vd. es el ordenador y desea recibir un dato del exterior, la hora por ejemplo, de la central telefónica de su ciudad. La central telefónica sería el canal. El teléfono sería el Interface y el número que Vd. ha de marcar para conseguir el contacto sería la vía (stream).

Cuatro de estas vías (streams) tiene el canal prefijado y no se pueden usar para otros canales.

Estas son:

vía (stream) Ø, vía (stream) 1 - los datos salen en la parte inferior de la pantalla y entran por medio del teclado.
vía (stream) 2 - los datos salen en la parte superior de la pantalla pero no pueden entrar datos por medio de esta vía (stream).
vía (stream) 3 - Los datos salen por la Impresora ZX pero no pueden entrar datos por esta vía (stream).

Comprobemos esta afirmaciones. Si teclea:

PRINT # 0; «PRUEBA»: PAUSE 0

Observará que se imprime en la parte inferior de la pantalla, donde normalmente el ordenador imprime los mensajes de error. El comando PAUSE 0 evita que se borre «PRUEBA» al contestar el ordenador con el mensaje OK, hasta que se pulse una tecla. Si ahora pulsa:

INPUT # 0; «VARIABLE»;N

Observará que tiene el efecto normal ya que siempre realiza la entrada de datos en esta parte inferior de la pantalla.

Compruebe ahora la vía(stream) 2 mediante:

PRINT # 2; «Prueba 2»

Y el mensaje «Prueba 2» aparecerá en la parte superior de la pantalla. Compruebe también que no puede entrar datos mediante:

INPUT # 2; «Variable»;n

Donde el ordenador devolverá un mensaje de error.

Todos los comandos que antes de conectar el Interface 1 producían una salida o entrada de datos en alguno de estos canales usaban estas vías (streams) automáticamente. Como hemos visto, PRINT usa la vía (stream) 2 y LPRINT la vía (stream) 3 automáticamente. Por lo tanto.

PRINT «Esto es una prueba»

es lo mismo que

PRINT # 2; «Esto es una prueba»

Pruébelo para comprobar que tienen el mismo efecto.

Aparte de este uso de las vías (streams) establecidos Vd. puede crear las suyas propias numerándolas del 4 al 15. Como adición a estas vías puede indicarle al ordenador el periférico que desea usar por medio de los especificadores de canal. Estos son:

«K» - para el teclado.
«S» - para la pantalla.
«P» - para la Impresora ZX.

Y algunos más que estudiaremos más adelante.

Así pues K, S y P son canales establecidos. Estos requieren que Vd. use una coma (,) como separador cuando cree una vía (stream). En el caso de otros canales puede usar una coma (,) o un punto y coma (;).

Por ejemplo, puede abrir la vía (stream) número 4 hacia la pantalla:

OPEN # 4, «S»

Y luego imprimir en ella a través de esta vía (stream):

PRINT # 4; «Escritura en pantalla»

En la creación y tratamiento de canales se utilizan unos nuevos comandos. Son los siguientes:

OPEN # - Abre una vía (stream).
INKEY$ # - Recoge de la vía (stream) un carácter.
INPUT # - Entra en el ordenador datos de la vía (stream).
PRINT # - Saca información hacia la vía (stream).
CLOSE # - Cierra la vía (stream).

Veremos a continuación estos comandos más detenidamente.

OPEN

Se utiliza para abrir una vía (stream) de información. Se debe usar de este modo:

OPEN # x;«m»;1;«Nom»

Donde:
x Indica el número de la vía (stream).
«m»;1; Indica que se opera con Microdrives, unidad 1.
«Nom» Indica el nombre de la vía (stream).

Una vez ejecutada esta operación tendremos una vía (stream) por la que podremos hacer circular la información y podremos estar usándola hasta que la cerremos.

Recuerde que en los casos que estamos estudiando hablamos siempre de los Microdrives, por lo que el número de la vía (stream) debe ser uno entre 4 y 15. Los otros, como hemos visto, están fijados para otros periféricos.

Esta operación ha realizado lo siguiente:

- crea una nueva vía (stream) y le asigna el número x.
- pone en marcha el Microdrive y consulta si hay algún archivo con el nombre especificado. Si no lo hay, el ordenador supone que se desea crear, por lo que esperará una sentencia para introducir datos. Si el archivo existe el ordenador esperará que usted lo lea, porque a un archivo grabado y cerrado no se le pueden añadir más datos.

Para evitar este inconveniente, en los ejemplos que vienen al final de esta sección, le incluyo un programa que le será util.
Como ya se indicó los canales Ø, 1, 2 y 3 están prefijados por el sistema por lo que los resultados de intentar abrirlos mediante OPEN es impredecible.

PRINT

Este comando se utiliza para colocar un dato dentro de un archivo después de haber abierto una vía (stream) con OPEN. Su modo de uso es:

PRINT # x;dato

Donde:
x Indica la vía (stream) de comunicación.
; Es un separador.
dato Es el dato a colocar en el archivo.

Dentro de las sentencias PRINT tienen una importancia vital los separadores. Como sabe, son tres:

- punto y coma (;), que no imprime nada.
- coma (,), que envía el cursor a la mitad de la línea.
- apóstrofe ('), que tiene el mismo efecto que ENTER.

Lo verá mas claro con este ejemplo:

COMANDO	ARCHIVO (# **5)**	
PRINT # 5;6;7	6 7	
PRINT # 5;6,7	6	7
PRINT # 5;6'7	6 7	

Por supuesto, como dato puede ser enviado cualquier número, carácter o variable, tal como se hace normalmente con el comando PRINT.

INPUT

Este comando toma de una vía (stream) el dato o datos que se le indican. Estos datos pueden ser números o caracteres. Su modo de uso es el siguiente:

INPUT # x;a;b;c ...

Donde:

x Indica el número de la vía (stream).

a;b;c Indica cualquier variable o conjunto de variables tanto numéricas como alfanuméricas.

Cuando se ejecuta este comando el ordenador va tomando bytes (o caracteres si se trata de cadenas alfanuméricas), hasta que encuentra un código 13. Este código 13 corresponde al retorno de carro (ENTER). Por esta razón es muy importante el modo como se crea un archivo para que luego pueda ser leído de forma sencilla y adecuada a la utilidad que se desee darle.

Debe poner atención cuando use este comando con variables alfanuméricas que contengan comillas ('') dentro de la cadena pues el ordenador entenderá al llegar a este carácter que la cadena ha finalizado.

Para evitar este efecto debe usar el comando con la instrucción LINE delante de la variable. Por ejemplo:

INPUT # 4; LINE a$

INKEY$

Este comando tiene un efecto parecido al anterior pero tomando de la vía (stream) un sólo carácter. Su forma de uso es:

INKEY$ # x

Y al igual que en el BASIC normal, debe ir acompañado de algún otro comando, asignándolo a una variable, imprimiéndolo, etc.

CLOSE

Cualquier vía (stream) que haya sido abierta debe ser cerrada, ya que

en caso contrario la información que contiene quedará mal archivada o perdida en su totalidad. La forma de hacerlo es:

CLOSE # x

Donde:

x Es el número del archivo a cerrar.

Durante esta operación el Microdrive coloca una marca de final de archivo y salva el último sector.

Recuerde que una vez cerrado un archivo en el cual esté escribiendo ya no podrá volver a introducir más datos en él.

Por supuesto no podrá cerrar las vías(streams) 0, 1, 2 y 3 pues están reservadas, como le indiqué, y permanecen siempre abiertas.

CLEAR

Este comando no se especifica claramente en el Manual de **Sinclair.** Su función es la de limpiar por completo el «área de datos de los canales», es decir, cierra todas las vías (streams) (las de 0 a 3 no las toca), y borra la información que pueda haber en los buffers de las vías (streams), al contrario que la instrucción CLOSE que los graba en el Microdrive. Su modo de uso es:

CLEAR #

MOVE

Esta instrucción le permite trasladar archivos de un canal a otro por medio de las vías (streams). Su modo de uso es:

MOVE # x1 TO # x2

Donde:

x1 Es la vía (stream) de donde salen los datos.
x2 Es la vía (stream) hacia la cual se envían los datos.

Cualquiera de ambas puede ser manipulada con un Microdrive. En ese caso x1 o x2 debe ser la expresión:

«m»;1;«Nombre»

Utilizando esta expresión no es necesario abrir ninguna vía (stream) con lo cual se ahorra memoria.
Las vías (streams) 1 a 3 deben ser indicadas por estos números y no se ejecutan si se utilizan los especificadores de canal: K, S o P.
Si una vía (stream) no está abierta o no encuentra el archivo en el Microdrive o existe un programa con el nombre del archivo, el ordenador devolverá el mensaje OUT OF RANGE.
Si se usa con archivos que no han sido cerrados en su creación puede haber serios problemas.
No es posible usar este comando con programas, sólo con archivos.
Algunos de los usos más útiles pueden ser:

MOVE «m»;1;«Nombre» TO # 2

o

MOVE «m»;1;«Nombre» TO # 3

que mostrarán el archivo «Nombre» en pantalla y en la impresora respectivamente.

ARCHIVOS

Consideraciones sobre Archivos

Siempre que trabaje con archivos tenga bien presentes estas consideraciones:

- El comando **PRINT** # puede contener información para varias vías (streams) a la vez. Por ejemplo:

PRINT # 4;«tu»; # 2;«yo»; # 10;«el»

El cual imprimirá «tu» en la via(stream) 4, «yo» en la pantalla y «el» en la vía(stream) 10.

- Si intenta crear un archivo con el mismo nombre de un programa ya existente, el ordenador devolverá el mensaje WRONG FILE TYPE pues no pueden existir dos archivos o programas con el mismo nombre.
- Algunos comandos de tratamiento de los colores de la pantalla (PAPER, INK, etc.) no tendrán efecto ejecutados detrás de una operación con vías (streams).

El modo de evitarlo es colocando delante de estos comandos la introducción: PRINT. Esto evitará el problema, aunque resulte raro de entender.

- Si intenta abrir una vía (stream) ya abierta el **Spectrum** devolverá el mensaje de error STREAM ALREADY OPEN.
- Si trata de leer un dato de un fichero que ha leído totalmente el ordenador devolverá el mensaje END OF FILE.
- Un catálogo puede ser enviado a un fichero mediante:

CAT # x,1

- Cuando se abre una vía (stream), ésta no se crea realmente en el Midrodrive, sino que ocupa en memoria una zona de 595 bytes con

un buffer de datos de 512 bytes. Estos bytes no son enviados al Microdrive hasta que se completan los 512 o se ejecuta un comando CLOSE a esa vía (stream). Es importante esta consideración, pues si se tiene en memoria un programa que la ocupe casi en su totalidad y no hay sitio para estos buffers de las vías (streams), al intentar abrir un vía (stream) el ordenador puede presentar el mensaje de error OUT OF MEMORY. Además, al crearse uno de estos buffers el programa en BASIC se desplaza hacia arriba en la memoria con lo cual, si contiene código máquina en líneas REM, éste quedará inutilizado al quedar en una posición de memoria distinta de la que fue concebido para funcionar.

CLS

Este es un comando extra, que no viene especificado en el Manual que entrega **Sinclair** con los Microdrives. Al usarlo, además de borrar la pantalla por completo coloca:

- la tinta **(INK)** en negro (Ø)
- el papel **(PAPER)** en blanco (7)
- el borde **(BORDER)** en blanco (7)
- el video inverso **(INVERSE)** en **«no»** (Ø)
- el brillo **(BRIGHT)** en **«no»** (Ø)
- el **OVER** en **«no»** (Ø)
- el **FLASH** en **«no»** (Ø)

O sea que realiza un borrado total de la pantalla y atributos relacionados con ella.

LIST

Este comando tampoco está en el Manual de **Sinclair**. Su función

es enviar el listado de un programa a través de una vía (stream). Resulta útil si se desea el listado para usarlo con un procesador de textos o para buscar en él una palabra.

Por supuesto puede ser enviado a un archivo en el Microdrive.

Su modo de uso es:

LIST # 4;n

Donde:

4 Es la vía (stream) que hemos abierto, hacia donde va el listado.
n Es la línea en la cual empieza el listado.

Programas con ficheros

Incluyo en este capítulo algunos programas de tratamiento de archivos que espero le sean de gran utilidad en su aprendizaje:

CREANDO UN ARCHIVO

Este programa crea un archivo con los números del 1 al 100.

```
10 REM Creación de un archivo
20 OPEN # 5;«m»;1;«Numeros»
30 FOR N=1 TO 100
40 PRINT # 5;N
50 NEXT N
60 CLOSE # 5
```

LEYENDO UN ARCHIVO

Este programa lee el archivo que hemos creado antes y lo imprime en pantalla.

```
10 REM Lectura de un archivo
20 OPEN # 5;«m»;1;«Numeros»
30 FOR N=1 TO 100
40 INPUT # 5;A
50 PRINT A;
60 NEXT N
70 CLOSE # 5
```

Sin embargo este programa puede ser sustituido por una sola instrucción:

```
MOVE «m»;1;«Numeros» TO # 2
```

AÑADIENDO DATOS

Como sabe, una vez cerrado un archivo no se le pueden añadir más datos. Este corto programa le introduce en el sistema para poder realizarlo.

```
10 REM Intercambio de archivos
20 OPEN # 5;«m»;1;«Nombre 2»
30 MOVE «m»;1;«Nombre 1» TO # 5
40 PRINT # 5;«Nuevos datos»
50 CLOSE # 5
60 ERASE «m»;1;«Nombre 1»
70 OPEN # 4;«m»;1;«Nombre 1»
80 MOVE «m»;1;«Nombre 2» TO # 4
90 ERASE «m»;1;«Nombre 2»
```

Este programa añade la frase «Nuevos datos» al archivo «Nombre 1» por medio del archivo «Nombre 2» que sirve de puente.

MEZCLANDO ARCHIVOS

Este programa mezcla dos archivos convirtiéndolos en uno solo.

```
10 REM Mezcla de archivos
20 OPEN # 5;«m»;1;«Tercero»
30 MOVE «m»;1;«Primero» TO # 5
40 MOVE «m»;1;«Segundo» TO # 5
50 CLOSE # 5
60 ERASE «m»;1;«Primero»
70 ERASE «m»;1;«Segundo»
```

BUSCANDO PALABRAS

Este programa busca en un archivo del Microdrive donde se haya colocado un listado de un programa mediante el comando LIST #, una palabra en concreto.

```
 10 REM Búsqueda de palabras.
 20 INPUT «Microdrive número?»; b
 30 INPUT «Nombre del archivo?»; b$
 40 OPEN # 5;«m»;b;b$
 50 INPUT «Palabra a buscar?»; LINE d$
 60 INPUT # 4; LINE a$
 70 FOR F=1 TO LEN a$ — LEN d$
 80 IF a$ (F TO F+LEN d$—1)=d$ THEN PRINT a$: PAUSE 50:
    GOTO 60
 90 NEXT F
100 GOTO 60.
```

Este programa finalizará con el mensaje de error END OF FILE.

LA RED LOCAL

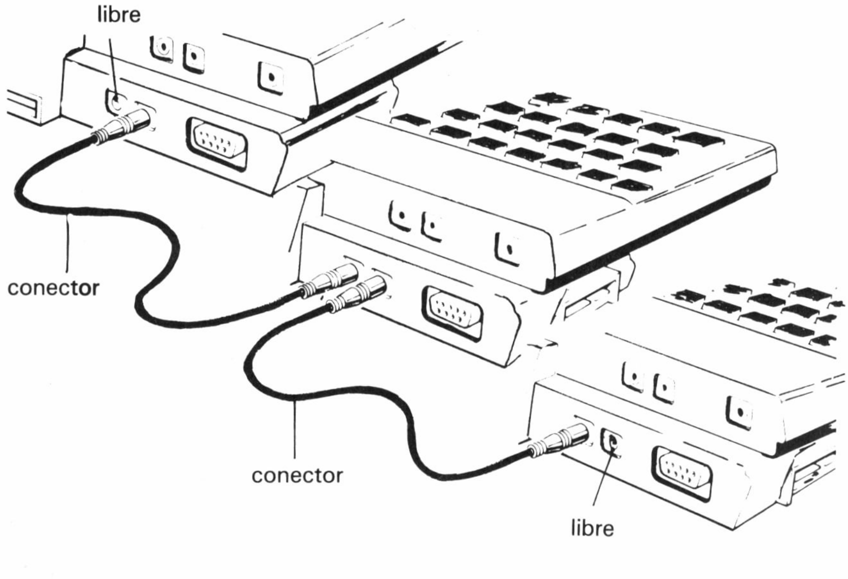
La Red Local

El área local de comunicaciones (network), permite conectar hasta 64 **Spectrums** y posibilita la comunicación entre ellos. A través de ella se pueden enviar tanto programas como datos, además de posibilitar el uso de un solo Microdrive por cualquiera de estos **Spectrums.**

Por medio del cable que venía empaquetado con el Interface 1, puede conectar dos **Spectrums** entre sí del modo que se indica en la figura inferior. Nunca debe cerrar el círculo de conexiones, es decir, siempre han de quedar dos **Spectrums** (los de los extremos) con un conector libre.

Asegúrese bien de no conectar o desconectar la alimentación de cualquier **Spectrum** cuando se esté realizando una comunicación, pues puede dañar al sistema.

La velocidad de transferencia por medio de esta red es 5K por segundo, la cual es mucho más rápida que la máxima velocidad que se puede conseguir con el Interface RS-232.



Utilizando la red local puede usar vías (streams) al igual que con el Microdrive, pero con el especificador de canal «N» (o «n»).

Antes de utilizarla debe asignarle a su **Spectrum** un número de estación de 1 a 64. Al conectar el **Spectrum** este número es el 1 y, si sólo se conectan dos **Spectrums** entre sí, no es necesario indicar el número de estación.

Para indicar este número de estación debe teclear:

FORMAT «n», número

PROGRAMAS

Por medio de la red local pueden ser enviados y recibidos programas y datos.

Para enviar un programa debe teclear:

SAVE * «n»; num

Donde:

***«n»** Indica que se desea usar la red local.
num Indica el número de estación hacia donde se envia el programa.

Observe que usando la red local no es necesario indicar el nombre del programa.

Al igual que con los Microdrives, pueden enviarse del mismo modo matrices, pantallas y código máquina colocando tras esta expresión las conocidas DATA, SCREEN$ o CODE según se trate.

Para recibir un programa debe pulsar:

LOAD * «n»; num

Donde:

***«n»** Indica que se desa usar la red local.
num Indica el número de estación de la cual se desea recibir.

Con este comando pueden usarse las mismas expresiones que con el anterior.

Al ejecutar estas operaciones al borde de la pantalla se activará de forma parecida (sic) a cuando se realiza SAVE.

Durante la ejecución de estos comandos el ordenador espera hasta que se ha realizado la operación completa, es decir, el que envía espera hasta que el otro reciba y viceversa.

Para verificar el programa recibido debe teclear:

VERIFY * «n»; num

y la estación que lo envió debe volver a pulsar:

SAVE * «num»; num

También puede usarse el comando MERGE, con el mismo efecto que con el cassette, del siguiente modo:

MERGE * «n»; num

Como habrá observado, al enviar o recibir programas no es necesario abrir ninguna vía (stream).

DATOS

Para enviar o recibir datos por medio de la red local se usan las vías (streams) de forma parecida a como se hacía con los Microdrives y con

las mismas instrucciones (OPEN, PRINT, INPUT, INKEY$, CLOSE). Para abrir una vía (stream) teclee:

OPEN # x;«n»; num

Donde:

x Indica el número de vía (stream).
;«n»; Indica que usaremos la red local.
num Indica el número de estación a la cual enviaremos datos o de la cual los recibiremos.

Esta operación crea un buffer similar a cuando se usaba con los Microdrives pero con diferente longitud (255 bytes).

Al abrir una vía (stream) debe decidir si será para enviar o para recibir datos pués una misma vía (stream) no puede ser usada para ambas operaciones simultáneamente. Vd. determina a cuál de ellas destinará la vía (stream) al realizar la primera operación con ella. Si utiliza el comando.

PRINT # x; dato

entonces será una vía (stream) para enviar los datos.

Por el contrario, si en la primera operación que realice usa el comando

INPUT # x; variable (para el dato)

el ordenador entenderá que se trata de una vía (stream) para recibir datos.

Igual que con los Microdrives, el comando

CLOSE # x

cierra la vía (stream).

Es importante este último comando ya que el **Spectrum** no enviará los datos hasta que esté lleno el buffer de la vía (stream) o hasta cerrarlo.

Así, si no se cierra la vía (stream), la información que contenga el buffer no será enviada y la estación que la esté esperando continuará haciéndolo siempre.

El efecto del comando INKEY$ # es parecido a cuando se utiliza con el teclado, es decir, devuelve la cadena vacía si en el momento de usarse no circula ninguna información por la red local.

ESTACION Ø

Normalmente, enviando o recibiendo datos o programas a otra estación la red local espera hasta que la operación esté completamente realizada en ambas estaciones.

Una facilidad de la red local es la posibilidad de enviar o recibir datos sin esta espera y a todas las estaciones a la vez. A esta facilidad se accede a través de la vía (stream) número Ø.

Cuando se envían datos o programas a través de la vía (stream) Ø, se envía a todos los **Spectrums** conectados a la red local pero no espera la conformidad de la información. Si alguna de las estaciones está esperando de esta vía (stream) recibirá la información pero las que no la esperen serán ignoradas.

Al enviar programas por esta vía (stream), las estaciones receptoras deben pulsar.

LOAD * «n»; Ø (y ENTER)

y la estación que envía el programa

SAVE * «n»; Ø (y ENTER)

Al enviar información a través de esta vía (stream) no podrá saber qué estaciones la han recibido pero en el caso del receptor podrá saber quién la envió pulsando.

PRINT PEEK 23759

pues en esta posición de memoria se almacena el número de estación que envía los datos.

JUEGO DE BARCOS

Juego de Barcos

Este programa muestra un ejemplo de uso de la red local para intercambio de datos.

El programa es el conocido juego de los barcos, para dos jugadores, y mediante la red local permite que cada jugador tenga en su ordenador el estado del juego en cada momento, y que los disparos y resultados sean enviados al adversario.

Al principio del programa se crean dos matrices de ocho por ocho elementos que van a contener las diversas posiciones de los barcos propios y de los del enemigo. Tal como se explica en el listado, la matriz A contiene las posiciones de los barcos propios y la matriz B la posición de los barcos enemigos.

En las líneas 30 a 50 inclusives, se borra la pantalla y se dibujan las cuadrículas para la presentación en pantalla de lo que acontece durante el juego.

En las líneas 60 a 460 inclusive se pide al jugador que entre las coordenadas de las posiciones donde desea colocar sus barcos y realiza la comprobación para que no pueda situar sus barcos en la misma casilla. Durante esta entrada de datos no se realiza la comprobación de si las coordenadas están mal entradas ni si coloca dos barcos distintos pegados entre si, para evitar alargar demasiado el programa, por lo que deberá asegurarse de que estos datos los entra correctamente.

En las líneas 480 a 550 inclusive se pregunta al jugador si las posiciones que hay en la pantalla son las correctas y si va a jugar primero. Obviamente uno de los dos jugadores ha de contestar que sí y otro que no. Al mismo tiempo, en la línea 510 se le asigna al **Spectrum** del jugador el número de estación para la red local, colocándolo a 1 pues sólo van a jugar dos ordenadores y es más cómodo usar como número de estación el 1.

La parte más importante de este programa comienza en la línea 1000, donde se realizan las diversas comunicaciones entre los dos ordenadores y se chequea la información recibida para decidir si el disparo ha dado en el blanco o en agua. Por esta razón se explica paso a paso cada rutina:

Líneas 1000 a 1060

En esta rutina se le pide al jugador que entre su disparo y se asigna

éste a una variable (w$). Luego se abre una vía (5) para enviar el disparo, cerrándola previamente por precaución. Tome nota de esto pues suele ser una buena forma de evitar errores. Después se envia a través de esta vía la variable, se cierra la vía y se toma nota del disparo asignando las coordenadas a las variables **«c»** y **«1»**.

Líneas 1110 a 1160

En esta rutina se abre otra vez la vía (5), cerrándola antes, y se espera de ella el resultado que debe devolver el otro jugador, asignándolo a la variable **«res»**. Luego se cierra la vía y se comprueba que el disparo no se haya efectuado anteriormente. A continuación se salta a la subrutina que presenta el disparo en pantalla y comprueba si se ha ganado la partida. En caso contrario se vuelve de la subrutina y sigue el programa.

Líneas 2000 a 2040

En esta rutina se abre la vía (5), cerrándola previamente y se espera del adversario que envíe el disparo. Una vez recibido se asignan las coordenadas de éste a las variables **«c»** y **«1»**.

Líneas 2050 a 2100

En esta rutina se analiza el disparo del enemigo y según haya dado en el blanco o no se le asigna un valor a la variable **«papel»** (para la impresión) y se abre la vía (5) cerrándola antes, para enviar el resultado del disparo. Luego se salta a la subrutina de impresión y se chequea si ha ganado la partida el adversario. En caso contrario se vuelve a la línea 1000 para continuar el ciclo de disparos entre ambos jugadores.

El resto del programa lo componen las subrutinas de impresión y comprobación de ganador que no son demasiado relevantes.

La estructura del programa permite que cada usuario pueda realizar las modificaciones que considere oportunas y no está demasiado pulido por este motivo, ya que su principal función es la de servir de ejemplo de uso de la red local.

```
   1 REM  *** BARQUITOS ***
   2 REM
   3 REM
   4 REM  § VICENTE MENDOZA
   5 REM
   6 REM
   7 REM  ZX-SPECTRUM  16K
   8 REM
   9 REM
  10 REM  VERSION PARA NETWORK
  11 REM
  12 REM
  20 DIM A(8,8): DIM B(8,8): DIM
 C(10)
  21 REM a=matriz propia
         b=matriz enemiga
  30 BORDER 0: PAPER 0: INK 7: C
LS : BRIGHT 1
  40 FOR N=16 TO 80 STEP 8: PLOT
 N,48: DRAW 0,64: NEXT N: FOR N=
48 TO 114 STEP 8: PLOT 16,N: DRA
W 64,0: NEXT N
  50 FOR N=168 TO 232 STEP 8: PL
-OT N,48: DRAW 0,64: NEXT N: FOR
N=48 TO 114 STEP 8: PLOT 168,N:
DRAW 64,0: NEXT N
  60 PRINT AT 2,10; PAPER 2;" BA
RQUITOS ";AT 6,2; PAPER 6; INK 2
;"SU FLOTA";AT 6,21;"ENEMIGOS"
  70 PRINT AT 16,2;"ABCDEFGH";AT
 16,21;"ABCDEFGH": FOR N=8 TO 15
: PRINT AT N,1;CHR$ (49+N-8);AT
N,20;CHR$ (49+N-8): NEXT N
  80 INK 0: PRINT AT 18,1; PAPER
 3;" "; PAPER 7;"Submarinos.";AT
 19,1; PAPER 4;" "; PAPER 7;"Tor
pederos.";AT 20,1; PAPER 6;" ";
PAPER 7;"Fragatas.  ": INK 7
 100 REM coloque sus barcos
 110 FOR n=1 TO 4
 120 PRINT AT 21,0;"Coordenadas
Submarino ";n: INPUT w$
```

```
 130 LET c=CODE (w$( TO 1))-96:
LET l=CODE (w$(2 TO 2))-48
 140 IF c<1 OR c>8 OR l<1 OR l>8
 THEN  GO TO 120
 150 IF a(l,c)=0 THEN  LET a(l,c
)=1: GO TO 170
 160 PRINT AT 21,0; FLASH 1;"
 CASILLA OCUPADA     ": BEEP 1,0:
 PAUSE 200: GO TO 120
 170 PRINT AT l+7,c+1; PAPER 3;
OVER 1;" "
 180 NEXT n
 200 FOR n=1 TO 3: FOR m=1 TO 2
 210 PRINT AT 21,0;"Coordenadas
Torpedero ";n: INPUT w$
 220 LET c=CODE (w$( TO 1))-96:
LET l=CODE (w$(2 TO 2))-48
 230 IF c<1 OR c>8 OR l<1 OR l>8
 THEN  GO TO 210
 240 IF a(l,c)=0 THEN  LET a(l,c
)=1: GO TO 260
 250 PRINT AT 21,0; FLASH 1;"
 CASILLA OCUPADA     ": BEEP 1,0:
 PAUSE 200: GO TO 210
 260 PRINT AT l+7,c+1; PAPER 4;
OVER 1;" "
 270 NEXT m: NEXT n
 300 FOR n=1 TO 2: FOR m=1 TO 3
 310 PRINT AT 21,0;"Coordenadas
Fragata ";n;"    ": INPUT w$
 320 LET c=CODE (w$( TO 1))-96:
LET l=CODE (w$(2 TO 2))-48
 330 IF c<1 OR c>8 OR l<1 OR l>8
 THEN  GO TO 310
 340 IF a(l,c)=0 THEN  LET a(l,c
)=1: GO TO 360
 350 PRINT AT 21,0; FLASH 1;"
 CASILLA OCUPADA     ": BEEP 1,0:
 PAUSE 200: GO TO 310
 360 PRINT AT l+7,c+1; PAPER 6;
OVER 1;" "
 370 NEXT m: NEXT n
```

```
 390 FOR m=1 TO 4
 400 PRINT AT 21,0;"Coordenadas
Portaaviones ": INPUT w$
 410 LET c=CODE (w$( TO 1))-96:
LET l=CODE (w$(2 TO 2))-48
 420 IF c<1 OR c>8 OR l<1 OR l>8
 THEN  GO TO 400
 430 IF a(l,c)=0 THEN  LET a(l,c
)=1: GO TO 450
 440 PRINT AT 21,0; FLASH 1;"
 CASILLA OCUPADA     ": BEEP 1,0:
 PAUSE 200: GO TO 400
 450 PRINT AT l+7,c+1; PAPER 1;
OVER 1;" "
 460 NEXT m
 480 PRINT AT 21,0;" "; PAPER 1;
" "; PAPER 7; INK 0;"Portaavion.
"; PAPER 0;"                     "
 500 INPUT "Posiciones correctas
? s/n ";w$: IF w$<>"s" AND w$<>"
S" THEN  GO TO 100
 510 FORMAT "n";1
 550 INPUT "Juega vd. primero? s
/n ";w$: IF w$(1)<>"s" AND w$(1)
<>"S" THEN  GO TO 2000
1000 REM Juegas tu
1010 INPUT "Disparo";w$
1020 LET w$=w$( TO 2)
1030 CLOSE #5: OPEN #5;"n";1
1040 PRINT #5;w$
1050 CLOSE #5
1060 LET c=CODE (w$( TO 1))-96:
LET l=CODE (w$(2 TO 2))-48
1100 REM Recibe el resultado
1110 CLOSE #5: OPEN #5;"n";1
1120 INPUT #5;res
1130 CLOSE #5
1140 IF b(l,c)=0 THEN  LET b(l,c
)=res
1150 LET papel=res
1160 GO SUB 8500
```

```
2000 REM Juega el enemigo
2010 CLOSE #5: OPEN #5;"n";1
2020 INPUT #5;w$
2030 CLOSE #5
2040 LET c=CODE (w$( TO 1))-96:
LET l=CODE (w$(2 TO 2))-48
2050 IF a(l,c)=1 THEN  LET res=2
: LET a(l,c)=2: LET papel=2: GO
TO 2070
2060 IF a(l,c)<>1 THEN  LET res=
5: LET papel=5
2070 CLOSE #5: OPEN #5;"n";1
2080 PRINT #5;res
2090 CLOSE #5
2100 GO SUB 8000: GO TO 1000
8000 REM Impresion propia
8010 PRINT AT l+7,c+1; PAPER pap
el; OVER 1;" "
8020 LET toc=0
8030 FOR n=1 TO 8: FOR m=1 TO 8
8040 IF a(n,m)=2 THEN  LET toc=t
oc+1
8050 NEXT m: NEXT m
8060 IF toc=28 THEN  GO TO 9500
8070 RETURN
8500 REM Impresion enemigo
8510 PRINT AT l+7,c+20; PAPER pa
pel; OVER 1;" "
8520 LET toc=0
8530 FOR n=1 TO 8: FOR m=1 TO 8
8540 IF b(n,m)=2 THEN  LET toc=t
oc+1
8550 NEXT m: NEXT m
8560 IF toc=28 THEN  GO TO 9000
8570 RETURN
9000 REM Ganas la partida
9010 CLS #: PRINT AT 10,10; FLAS
H 1;"HAS PERDIDO": STOP
9500 REM Pierdes la partida
9510 CLS #: PRINT AT 10,10; FLAS
H 1;"HAS PERDIDO": STOP
```

EL INTERFACE RS-232

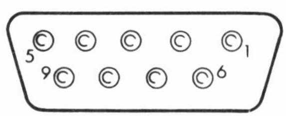
El Interface RS-232

El Interface RS-232 permite al **Spectrum** comunicarse con multitud de periféricos e incluso otros ordenadores.

Este Interface suele incluirse como estandar en la mayoría de ellos.

Para realizar la conexión debidamente deberá consultar detenidamente el manual del periférico a conectar y hacerlo de acuerdo a estas especificaciones.

1. **No conectar.**
2. **TX (entrada).**
3. **RX (salida).**
4. **DTR (entrada) debe ser alto cuando esté libre.**
5. **CTS (salida) debe ser alto cuando esté libre.**
6. **No conectar.**
7. **Conexión a masa.**
8. **No conectar.**
9. **+ 9 v.**



Después debe ajustar el periférico según estos condicionantes:

- Según la aplicación, el salto automático de línea en OFF, pues el **Spectrum** envía en algunas ocasiones el código de salto de línea (se verá más adelante).
- La paridad en OFF.
- El número de bits, en 8.
- El número de bits de stop en 1.

En cuanto a la velocidad de transferencia (en baudios) es seleccionable desde el **Spectrum** (por software) y puede ser cualquiera de las estandar:

50, 110, 300, 600, 1200, 2400, 4800, 9600 y 19200
(en este caso: baudios/11 = bytes por segundo)

Lo ideal es seleccionar la máxima velocidad a la que pueda funcionar el periférico que va a conectar.
La forma de seleccionar estos baudios en el **Spectrum** es mediante el comando:

FORMAT «t»; baudios

o

FORMAT «b»; baudios

«t» y **«b»** son, en el caso del Interface RS-232, los especificadores de canal. Ambos son diferentes y pueden usarse simultáneamente.

CANAL T

Este canal se usa para enviar a través del RS-232 cualquier texto. En relación con los códigos de los caracteres (ver apéndice A del manual del Spectrum), se comporta del siguiente modo:

CODIGOS	FUNCIONAMIENTO
0-12	Códigos de control. Los ignora.
13	Retorno de carro. Envía un retorno de carro (13) y un salto de línea (10).
14-31	Códigos de control. Los ignora.
32-127	Códigos ASCII. Los envía sin modificar.
128-164	Códigos gráficos. Envía un interrogante (?)
165-255	Comandos del **Spectrum**. Los envía carácter a carácter.

Al ejecutar un comando **INPUT** o **INKEY$** por este canal el **Spectrum** espera 7 bits, ignorando el octavo.

Este canal es idóneo para el uso de impresoras con **Interface RS-232** pués una vez abierto un stream mediante:

OPEN # 3; «t»

los comandos **LLIST** y **LPRINT** se ejecutan directamente hacia este canal.

Como ha visto, pueden abrirse streams en el canal **«t»** y usar todos los comandos que usábamos con archivos, es decir:

OPEN # x; «t»
PRINT # x; dato
INPUT # x; dato
INKEY$ # x
CLOSE # x

Pero si tiene que enviar un código de control a una impresora o necesita que el **Spectrum** tenga en cuenta los 8 bits que entran por el **RS-232**, tendrá que usar otro canal para ello: el **«b»** (binario).

CANAL B

Como hemos dicho, este canal permite enviar o recibir cualquier código de 8 bits, pues no realiza ningún tipo de conversión de códigos. Además este canal le permite enviar o recibir programas por medio del **RS-232**.

Para ello debe usar primero el comando:

FORMAT «b»; baudios

Luego si quiere enviar un programa pulse:

SAVE * «b»

y para recibirlo:

LOAD * «b»

Las demás expresiones usuales **(SCREEN$, CODE, LINE)**, están permitidas del siguiente modo:

SAVE * «b»; SCREEN$

o

SAVE * «b»; LINE n

Como en el canal «t», pueden ser utilizados los comandos:
OPEN # x; «b»
PRINT # x; dato
INPUT # x; dato
INKEY$ # x;
CLOSE # x;

Consideraciones

- No es posible conectar dos periféricos a la vez vía **RS-232** pero sí

que diferentes canales envíen o reciban información a través de él. Tampoco es posible que diferentes canales tengan diferente velocidad en baudios.

• Al seleccionar una velocidad en baudios, ésta se ajustará al número inferior más próximo de los estandar, es decir, si por ejemplo intenta utilizar una velocidad de 70 baudios, el **Spectrum** la ajustará a 50 por ser la más próxima dentro de las estandar.

• Cuando envía datos vía **RS-232** el borde de la pantalla se vuelve negro si el periférico da al **Spectrum** la señal de ocupado y el ordenador espera la señal de libre para enviar la información.

• Cuando cierra (CLOSE) una vía (stream) que operaba con el **RS-232**, el **Spectrum** envía el código 13 hacia ésta. Esto no se efectúa si utiliza el comando CLEAR.

• Si olvida la velocidad en baudios que está usando, puede conocerla mediante:

```
PRINT INT (3500000/((PEEK 23747+256*PEEK 23748+2)*26))
```

Ejemplos de uso del RS-232

Supongamos que quiere tener por impresora un listado doble alto,

que el código doble alto de su impresora es el 14 y funciona a una velocidad de 1200 baudios. Para poder hacerlo teclee este programa:

```
10 REM Listado doble alto
20 FORMAT «t»; 1200
30 FORMAT «b»; 1200
40 OPEN # 5; «t»
50 OPEN # 10; «b»
60 PRINT # 10; CHR$ 14;
70 LIST # 5
80 CLOSE # 5
90 CLOSE # 10
```

Se puede obtener una copia de la pantalla por impresora mediante este programa (en baja resolución):

```
 10 REM Copy
 20 FORMAT «t»; baudios
 30 OPEN # 5; «t»
 40 FOR N=0 TO 21
 50 FOR M=0 TO 31
 60 LET B$=SCREEN$ (N,M)
 70 IF B$=«» THEN LET B$=« »
 80 PRINT # 5; B$;
 90 NEXT M
100 PRINT # 5
110 NEXT N
```

Otra de las ventajas que incorpora el **RS-232** si lo usa con una impresora es la posibilidad de usar los comandos **LLIST** y **LPRINT** directamente.

Para ello habrá de formatear primero el canal **«t»** a la velocidad en baudios de su impresora y luego abrir la vía (stream) número 3 (SIEMPRE LA 3) hacia ese canal, o sea:

```
FORMAT «t»; baudios        (ENTER)
OPEN # 3; «t»              (ENTER)
```

y el sistema estará listo para aceptar estos comandos:

```
LLIST
LPRINT
```

LA MEMORIA

Estado de la Memoria

Si ha leído atentamente las páginas anteriores habrá observado que a partir de la conexión del Interface 1 al **Spectrum** la memoria de éste ha sufrido grandes modificaciones.

La más destacable se indicó en un capítulo anterior y consiste en los movimientos arriba y abajo de la memoria del programa BASIC cuando se usa el Microdrive. Como apuntamos en varios capítulos ello se debe a los buffers que necesitan los canales y a que, cada vez que se usa el Microdrive, el **Spectrum** se crea en memoria un mapa completo de él, sin embargo, el usuario no notará nada anómalo si el programa que ejecuta es BASIC, sin POKES o PEEKS en la propia zona de programa (por ejemplo, datos o código máquina en líneas REM).

Para evitarle problemas con la memoria se le incluyó un programa que le informará del espacio que ocupa cada cosa y la memoria que tiene libre en cualquier momento. Si desea usarlo de la mejor manera, cambie los números de línea e introdúzcalo como una subrutina dentro de su propio programa y acceda a ella cada vez que lo crea necesario.

```
   1 REM  ESTADO DE LA MEMORIA
   2 REM
   3 REM
   4 REM   $ VICENTE MENDOZA
   5 REM
   6 REM
   7 REM
   8 REM
   9 REM
  10 DEF FN m(n)=PEEK n+256*PEEK
 (n+1)
  20 PAPER 7: INK 0: BORDER 7: C
LS
  30 DRAW 255,0: DRAW 0,175: DRA
W -255,0: DRAW 0,-175
  40 PRINT AT 1,5; PAPER 2; INK
6;"ESTADO DE LA MEMORIA"
  50 PRINT AT 4,1;"RAM (total)";
AT 4,18;(FN m(23732)-16383)/1024
;AT 4,23;" Kbytes"
  60 PRINT AT 6,1;"Mapas MICRODR
IVE";AT 6,18;FN m(23631)-23734;A
T 6,23;" bytes"
  70 PRINT AT 8,1;"Programa BASI
C";AT 8,18;FN m(23627)-FN m(2363
5);AT 8,23;" bytes"
  80 PRINT AT 10,1;"Variables";A
T 10,18;FN m(23641)-FN m(23627)-
1;AT 10,23;" bytes"
  90 PRINT AT 12,1;"Memoria libr
e";AT 12,18;FN m(23730)-FN m(236
41);AT 12,23;" bytes"
 100 PRINT AT 16,1;"Graficos def
inibles:"
 110 REM Las mayusculas de la
         linea 120 deben ser
         introducidas en modo
         grafico.
         (CAPS SHIFT+9)
 120 PRINT AT 18,5;"ABCDEFGHIJKL
MNOPQRSTU"
```

El programa le presentará en pantalla:

```
     ESTADO DE LA MEMORIA

RAM (total)          47.00 Kbytes
Mapas MICRODRIVE     0     bytes
Programa BASIC       1148  bytes
Variables            0     bytes
Memoria libre        39444 bytes

Graficos definibles:
     ABCDEFGHIJKLMNOPQRSTU
```

Veamos ahora cómo conseguir acceder al mapa del Microdrive que se crea en memoria cuando lo usamos.

Vamos a crear un archivo en el Microdrive donde tendremos los números del 1 al 2000 mediante este programa:

```
10 REM Números
20 OPEN # 5;«m»;1;«Números»
30 FOR N=1 TO 2000
40 PRINT # 5; N;
50 NEXT N
60 CLOSE # 5
```

Una vez ejecutado el programa pulse NEW y teclee este programa:

```
10 REM Registros-Sectores
20 OPEN # 5;«m»;1;«Números»
30 PRINT «Registro»; PEEK 23912; «en el Sector»; PEEK 23885
40 POKE 23856,3
50 GOTO 30
```

Este programa se detendrá con el mensaje de error **END OF FILE**.

Explicaré ahora qué es lo que realiza el programa.

En primer lugar ha debido pulsar **NEW** con lo cual se borra cualquier contenido de la memoria incluidas las vías (streams) que pudiera haber. Así el buffer de la primera vía (stream) que se utilice se creará a partir de la dirección de memoria 23844.

Esto es necesario para poder calcular los datos de esta vía (stream), pues de no hacerlo hubiéramos tenido los datos de la primera vía (stream) que hubiera en memoria.

Luego hemos abierto una vía (stream) para la lectura de los datos. Al realizar esta operación, dentro del buffer se han creado una serie de variables, que hemos consultado con **PEEK 23912** y **PEEK 23885.**

Veamos más a fondo el contenido de estos buffers.

Los Buffers de los canales del Microdrive

Cada vez que abre una vía (stream) con el comando OPEN se crea en memoria un fichero (llamado CHANNEL) dentro de la zona llamada CHANS en el manual de **Sinclair** (capítulo 24). Este fichero tiene una longitud de 595 bytes de los cuales 512 son un buffer donde se almacena la información que maneja la vía (stream).

El resto de bytes los ocupan unas variables que contienen información sobre la vía (stream). Estas variables son las siguientes:

BYTES	NOMBRE	CONTENIDO
0-1		Dirección 8. (ØØØ8h). Dirección de la rutina en **INPUT.**
2-3		Dirección 8. (ØØØ8h). Dirección de la rutina en **OUTPUT.**

BYTES	NOMBRE	CONTENIDO
4		Carácter **«M»**.
5-6		Dirección de la subrutina de salida en **ROM «oculta»**.
7-8		Dirección de la subrutina de entrada en la **ROM «oculta»**.
9-10		Contiene el número «595». (Longitud del buffer de la vía (stream).
11-12	CHBYTE	Contiene el número del siguiente byte a añadir o a leer del buffer de datos en el rango de Ø a 512 inclusive.
13	CHREC	Contiene el número de registro con el que está trabajando, dentro del rango de Ø a 255.
14-23	CHNAME	Contiene los 10 bytes que forman el nombre de la vía (stream).
24	CHFLAG	Contiene un señalizador de lectura/escritura, es decir el bit Ø está a 1 si la vía (stream) está abierta para escritura y está a Ø, cuando la vía (stream) se usa para lectura de datos. Los bits 1 a 7 no se usan y están todos a 1.
25	CHDRIV	Contiene el número de Microdrive que se está usando, dentro del rango de 1 a 8.
26-27	CHMAP	Dirección del **MAPA** de este Microdrive.
28-39	CHMAP	Estos 27 bytes son la zona de trabajo donde se va elaborando una cabecera que se grabará en el Microdrive conteniendo un resumen de los datos del archivo (nombre, longitud, etc.).
40	HDFLAG	Contiene un señalizador que usa sólo el bit Ø para señalizar la cabecera.
41	HDNUMB	Contiene el número del sector que se ha grabado o leído.

BYTES	NOMBRE	CONTENIDO
42-43		NO USADOS.
44-53	HDNAME	Contiene el nombre del cartucho que se está usando.
54	HDCHK	Sumatorio de comprobación de la cabecera.
55-56		12 bytes conteniendo el preámbulo del bloque de datos. Marca el comienzo de la zona de manejo de los datos.
67	RECFLG	Contiene un indicador: bit Ø a Ø - no es cabecera. bit 1 a Ø - no es final del fichero. bit 1 a 1 - es final del fichero. bit 2 a Ø - es archivo de escritura. bit 2 a 1 - no es archivo de escritura.
68	RECNUM	Número de registro dentro del rango de Ø a 255.
69-70	RECLEN	Contiene un número que indica la longitud del registro dentro del rango de Ø a 512.
71-80	RECNAM	Contiene los caracteres que forman el nombre del archivo.
81	DESCHK	Contiene un sumatorio de los 14 bytes anteriores.
82-593	CHDATA	Contiene el buffer con los 512 bytes de datos.
594	DCHK	Contiene un sumatorio para la comprobación de los 512 bytes anteriores (datos).

Mapas del Microdrive

Como habrá observado, en la explicación de algunas de estas variables se habla de los mapas de los Microdrives. Estos se crean en memoria para que el ordenador conozca cuáles y cuántos sectores están libres, ocupados o inutilizados.

En principio el cartucho del Microdrive contiene 180 sectores aunque el software del sistema (ROM) está preparado para manejar hasta 256 sectores.

Por cada vía (stream) que abra para usar con el Microdrive se crea en la memoria un mapa del cartucho conteniendo los sectores libres y los que no se pueden usar. Vamos a echar una mirada a estos mapas para comprender mejor el funcionamiento del Microdrive.

El mapa contiene un total de 32 bytes. Cada uno de estos bytes contiene el estado de 8 sectores del Microdrive (uno por cada bit).

Si el sector contiene datos o es inutilizable el bit será 1 y si está libre el bit será Ø.

El siguiente programa le mostrará el mapa completo de un cartucho.

Para que funcione correctamente puede primero formatear el cartucho mediante:

FORMAT «m»;1;«cartucho 1»

Debe pulsar **NEW** seguido de **ENTER** para eliminar cualquier vía (stream) que hubiera en memoria.

Luego teclee este programa:

```
1Ø REM Mapa del Microdrive
2Ø OPEN # 5;«m»;1;«mapa»
3Ø FOR N=1 TO 32
4Ø LET Y= PEEK (23792+N)
5Ø FOR X=1 TO 8
6Ø PRINT Y/2 <  > INT (Y/2)
7Ø LET Y=INT (Y/2)
8Ø NEXT X
9Ø NEXT N
```

Y hágalo funcionar mediante **RUN**.

Este programa abre una vía (stream) para que se cree en memoria el mapa del Microdrive y luego consulta los 32 bytes que la componen bit a bit, imprimiendo en pantalla un Ø si el sector está libre o un 1 si está inutilizado.

Presentará en la pantalla algo parecido a esto:

```
Ø1ØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØ
ØØØØØØØØØØØØØØØØØØØ1111111111111
11111111111111111111111111111111
11111111111111111111111111111111
```

Donde puede observar que el sector 1 está inutilizado así como los sectores del 181 al 256, con lo cual quedan 199 sectores libres que dan un total de capacidad de 90 Kbytes (512 bytes por sector).

Los canales de la red local

Al igual que con los Microdrives, cuando se abre una vía (stream) direccionada a la red local se crea en memoria un buffer (zona de memoria para trabajo del ordenador), conteniendo los datos a enviar o recibir y una serie de variables que usa el ordenador.

Esta área de información tiene una longitud de 276 bytes y contiene un buffer de datos de 255 bytes además de estas variables:

BYTE	*NOMBRE*	*CONTENIDO*
0-1		Dirección 8. (ØØØ8h)
2-3		Dirección 8. (ØØØ8h)
4		Contiene el caracter «N».

5-6		Dirección de la subrutina de salida en ROM.
7-8		Dirección de la subrutina de entrada en ROM.
9-10		Contiene el número «256». (Longitud del buffer de la vía (stream).
11	NCIRIS	Contiene el número de la estación de destino.
12	NCSELF	Contiene el número de estación propio.
13-14	NCNUMB	Número de bloque, dentro del rango de 0 a 65535.
15	NCTYPE	Contiene un código para identificar el tipo de dato. Es 0 si son datos y 1 si ha de dar el mensaje de error **END OF FILE**.
16	NCOBL	Contiene la longitud del buffer de datos a enviar. Contendrá un 0 cuando se esté recibiendo.
17	NCDCS	Sumatorio de comprobación para los datos.
18	NCHCS	Sumatorio de comprobación de la cabecera.
19	NCCUR	Posición del último carácter recibido en el buffer.
20	NCIBL	Contiene el número de bytes que deben ser leídos del buffer de datos.
21-275	NCB	Contiene el buffer de datos. (Longitud: 255 bytes).

Las variables del sistema

Además de las variables del sistema que se crean al usar el Microdriver o la red local en las vías (streams) de éstos, con el Interface 1 se añaden más variables del sistema a las que constan en el Capítulo 25 del manual de **Sinclair.**

Estas nuevas variables son:

DIRECCION	NOMBRE	CONTENIDO
23734	FLAGS 3	Indicadores. Si los bits están a 1 indican lo siguiente: *bit 0*-Cuando se usa la Extensión del BASIC que contiene el Interface-1. *bit 1*-Durante la ejecución del comando **CLEAR**. *bit 2*-Cuando la variable **ERRSP** (23613), está alterada. *bit 3*-Cuando está funcionando la red local. *bit 4*-Cuando se ejecuta el comando **LOAD.** *bit 5*-Cuando se ejecuta el comando **SAVE.** *bit 6*-Cuando se ejecuta el comando **MERGE.** *bit 7*-Cuando se ejecuta el comando **VERIFY.**

DIRECCION	NOMBRE	CONTENIDO
23735	VECTOR	Contiene la dirección de salto cuando la ROM del Interface 1 da un error. Normalmente: 23735 contiene: 240 23736 contiene: 1

Se utiliza para ampliar el juego de instrucciones por el usuario.

23737	SBRT	Rutina usada por la nueva ROM para llamar a las subrutinas de la ROM del **Spectrum**, en la forma: LD HL, valor CALL subrutina LD (5CBAH), HL RET

23747	BAUD	Contiene un número de dos bytes que representan la velocidad en baudios del RS-232.

La fórmula usada para el cálculo de esta velocidad es:

baudios: (3500000 / (26 * velocidad en baudios)) - 2

Puede usarse esta dirección para hacer que el **Spectrum** funcione a una velocidad en baudios diferente a las indicadas como estandar.

DIRECCION	NOMBRE	CONTENIDO
23749	NTSTAT	Contiene el número de estación de la red local que le corresponde al propio **Spectrum**, dentro del rango de 1 a 64.
23750	IOBORD	Contiene el número del color del borde de la pantalla durante las operaciones de entrada y salida.

Si realiza un POKE en esta dirección entre Ø y 7 el borde cambiará al color que le indique.

Si no desea el efecto de flash durante las operaciones use:

POKE 23750, INT (PEEK 23624/8)

23751	SER-FL	Contiene dos bytes que corresponden a la zona de trabajo del **RS-232.** El primero son indicadores y el segundo es el carácter recibido.
23753	SECTOR	Espacio de trabajo usado por los Microdrives, normalmente para contar los sectores del cartucho.

DIRECCION	NOMBRE	CONTENIDO
23755	CHADD_	Posición de almacenamiento temporal de la variable CH_ADD, mientras chequea la sintaxis de los nuevos comandos en una línea.
23757	NTRESP	Almacenamiento del código de respuesta de la red local.
23758	NIDEST	Principio de buffer de la red local. Contiene el número de la estación de destino.
23759	NTSRCE	Contiene el número de la estación fuente.
23760	NTNUMB	Número de bloque de la red local, dentro del rango de Ø a 65535.
23762	NTTYPE	Código del tipo de cabecera de la red local. Contiene un Ø si son datos o 1 si ha llegado al final del archivo.

DIRECCION	NOMBRE	CONTENIDO
23763	NETLEN	Contiene la longitud del bloque de datos de la red local dentro del rango de Ø a 255.
23764	NTDCS	Sumatorio de comprobación del bloque de datos de la red local.
23765	NTHCS	Sumatorio de comprobación de la cabecera de la red local.
23766	D_STR1	Principio de los primeros 8 bytes especificadores de canal. Estos bytes indican el número de unidad de Microdrive o el número de la estación de destino de la red local o la velocidad en baudios del **RS-232.**
23768	S_STR1	Número de vía (stream), dentro del rango de Ø a 15.
23769	L_STR1	Especificador de canal. **(«M», «N», «T»** o **«B»)**

DIRECCION	NOMBRE	CONTENIDO
23770	N_STR1	Longitud del nombre del archivo.
23772	T_STR1	Principio del nombre del archivo. Normalmente en el espacio de trabajo.
23774	D_STR2	Principio de los segundos 8 bytes especificadores de archivo.
23782	HD_ØØ	Principio de la zona de trabajo para SAVE, LOAD, VERIFY Y MERGE. Contiene el código del tipo de archivo. Los códigos son: 0-Si es un programa. 1-Si es una matriz numérica. 2-Si es una matriz alfanumérica. 3-Si son bytes.
23783	HD_OB	Longitud de los datos, dentro del rango de Ø a 65535.
23785	HDD	Principio de los datos, dentro del rango de Ø a 65535.

DIRECCION	NOMBRE	CONTENIDO
23787	HD_OF	Longitud del programa, dentro del rango de Ø a 65535. (Puede contener el nombre de la matriz a cargar). Usada también por el código de llamada «32H».
23789	HD_11	Número de línea donde comienza la autoejecución de un programa.
23791	COPIES	Número de copias a realizar por el comando SAVE. Se coloca a 1 después de ejecutado.

Alterando el contenido de esta posición de memoria puede hacer que automáticamente se realicen varias copias del mismo programa en un cartucho del Microdrive, con lo que verá reducido considerablemente el tiempo de acceso a él, aunque obviamente el programa ocupará más memoria en el cartucho.

Para ello pulse antes del comando SAVE:

POKE 23791, NUM

Donde NUM es el número de copias a realizar.

Para eliminar un programa grabado **n** veces con **POKE 23791,n: SAVE *«m»;1;«prog»**, hay que hacer **n** veces **ERASE «m»;1;«prog»**.

El Interface 1 desde Código Máquina

Este capítulo está escrito para aquellos lectores que deseen usar el Interface 1 desde Código Máquina. Es obvio que para la comprensión de lo que se va a exponer son necesarios unos mínimos conocimientos de código máquina del microprocesador **Z-80 A**, y del funcionamiento interno del **Spectrum**.

La ROM oculta

Como ya indicamos en otro capítulo, el Interface 1 posee una ROM de 8 K en adición a la que posee el **Spectrum** de 16 K, para poder ejecutar los nuevos comandos y las distintas tareas que conlleva la utilización de los Microdrives, La Red Local y el Interface RS-232.

Esta ROM está paginada desde la dirección de memoria 0000 h (hexadecimal) hasta la dirección 1FFF h (hexadecimal). Como puede observar, estas direcciones también son usadas por la ROM de 16 K del **Spectrum**, la que soporta el Intérprete BASIC. Por este motivo se utiliza un peculiar sistema de paginado para «conectar» o «desconectar» la nueva ROM que conviene explicar.

En el **Spectrum**, la operación Restar 0008 h. se usa para identificar los errores. Estos errores están numerados de FF h. a 1A h. en la ROM del **Spectrum**. Se usa seguido de una instrucción DEF B xx, donde «xx» indica el número de error.

Con la incorporación del Interface 1 si se intenta utilizar uno de los nuevos comandos desde BASIC, la ROM del **Spectrum** fuerza un error usando la instrucción RST 8, con lo cual se activa la ROM «oculta» del Interface 1, la cual chequea el nuevo comando y lo ejecuta, si es correcto, volviendo a «conectarse» la ROM del **Spectrum** una vez ejecutado el comando.

Una vez en la ROM «oculta», ésta utiliza un sistema semejante para ejecutar cada una de las subrutinas necesarias para los nuevos comandos, es decir, utiliza el comando RST 8, seguido de DEF B xx, con un número que en este caso no indica el error sino la subrutina de la nueva ROM que ha de ejecutarse. Este número está dentro del rango de 1B h. a 32 h. y son los llamados por los ingleses: «hook codes».

Así pues, cada «hook code» se usa para llamar a una distinta subrutina dentro de la ROM oculta del siguiente modo:

```
RST 8
DEF B «hook code»
```

Pero el usuario no puede trabajar libremente en la ROM «oculta» ya que se desconecta cada vez que se acaba de usar.

Por si le interesa desensamblarla, el mejor sistema es usando este programa con un Microdrive:

```
10 REM Salvar la ROM como CODE al Microdrive
20 CLEAR 32767
30 SAVE * «M»;1;«ROM-8K» CODE 0,8192
40 LOAD * «M»;1;«ROM-8K» CODE 32768
```

Recuerde restar a todas las direcciones absolutas 8000 h. para obtener la dirección real.

A continuación estudiaremos cada «hook code» por separado, pero en cualquier caso tenga en cuenta las siguientes consideraciones:

—Los registros no son conservados, por lo que deberá guardar el valor que contengan al usar algún «hook code», especialmente los contadores y, en algunos casos, debe conservar la dirección de retorno al BASIC.

—Durante todas las transferencias de Entrada/Salida, las interrupciones están desactivadas, algunos «hook code» las desactivan y algunos otros requieren que al entrar en ellos estén desactivadas.

—El «hook code» 31 h. crea las nuevas variables del sistema (consultar el capítulo correspondiente) y debe ser usado cuando éstas sean necesarias, para entrar en las subrutinas de la ROM «oculta».

—El «hook code» 32 h. está reservado por **Sinclair**, aunque se podrá observar que es de gran ayuda para entrar en la nueva ROM sin activar ninguna subrutina.

Los «Hook codes»

En las explicaciones que se van a dar a continuación de cada uno de estos «hook codes» se intenta dar una explicación de la utilidad que tienen y el modo de usarlos.

A pesar de ello puede haber algunos que respondan de una manera poco usual. Es debido a que es enormemente complicado introducirse a fondo en el funcionamiento de la nueva ROM.

A esta nueva ROM del Interface 1 se ha optado por llamarla la ROM «oculta» debido al sistema de activarla y desactivarla que usa el sistema.

Las localizaciones que se indican para cada subrutina se refieren a su dirección en la ROM «oculta».

Observe que se está hablando de una misma cosa cuando se indica «hook code» y subrutinas, siempre que éstas se refieran a la ROM «oculta».

HOOK CODE 1B h. - ENTRADA DEL TECLADO

Esta subrutina espera hasta que se pulsa una tecla (en el teclado, por supuesto). Una vez pulsada, el código del carácter es devuelto en el

registro A. Las teclas que no dan un código de carácter son ignoradas. Su modo de uso es:

RST 8
DEF B 1B h.

Necesita: NADA.

Devuelve: En el registro A el código del carácter. Las interrupciones activadas.

Usa los registros: AF, BC, DE, HL.

Localización: 19D9 h.

HOOK CODE 1C h. - IMPRESION EN PANTALLA

Esta subrutina presenta en pantalla el carácter cuyo código esté cargado en el registro A. Cabe destacar que el scroll se ejecuta automáticamente. Su modo de uso es:

RST 8
DEF B 1C h.

Necesita: En el registro A el código del carácter. Las interrupciones activadas.

Devuelve: NADA.

Usa los registros: AF, BC, DE, HL, A'F', B'C', D'E'.

Localización: 19E9 h.

HOOK CODE 1D h. - ENTRADA DE DATOS POR RS-232.

Esta subrutina acepta bytes de la entrada RS-232. La velocidad de recepción está indicada por la variable del sistema BAUD y el color del

borde por la variable del sistema IOBORD. Es muy conveniente «limpiar» las variables del sistema SER FL antes de usar esta subrutina. Su modo de uso es:

RST 8
DEF B 1D h.

Necesita: NADA.

Devuelve: El flag (banderín) de carry (acarreo) activado si se ha leido un byte.

Usa los registros: AF, BC, DE, HL.

Localización: OB81 h.

HOOK CODE 1E h. - SALIDA DE DATOS POR RS-232.

Esta subrutina envía bytes a través del RS-232. Al igual que la anterior, la variable del sistema BAUD determina la velocidad de transferencia y la variable IOBORD determina el color del borde. Como es lógico, los bytes no se envían hasta que DTR mantiene una señal alta. Su modo de uso es:

RST 8
DEF B 1E h.

Necesita: En el registro A el código del byte a enviar.

Devuelve: Las interrupciones activadas.

Usa los registros: AF, BC, DE, HL.

Localización: 0C5A h.

Las dos subrutinas anteriores permiten enviar o recibir datos a través del canal «b» del RS-232. Para poder usar el canal «t» para enviar datos a través de él, puede usar el «hook code» 32 h. que veremos más expli-

cado en las próximas páginas, para luego hacer un CALL 0C3C h., con el código del carácter a enviar, cargado en el registro A.

HOOK CODE 1F h. - IMPRESION EN LA ZX-PRINTER

Esta subrutina imprime un carácter en la ZX-Printer. Su modo de uso:

RST 8
DEF B 1F h.

Necesita: En el registro A el código del caracter. Las interrupciones activadas.

Devuelve: NADA.

Usa los registros: AF, BC, DE, HL, A'F', B'C', D'E'.

Localización: 19FC h.

HOOK CODE 20 h. - TESTEO DEL TECLADO

Esta subritina testea el teclado para comprobar si se ha pulsado una tecla, incluidas las teclas de SHIFT. Si se ha pulsado o no una tecla viene indicado por el flag (banderín) de cero. Su modo de uso es:

RST 8
DEF B 20 h.

Necesita: NADA.

Devuelve: NZ si se ha pulsado una tecla.

Usa los registros: AF, BC, DE, HL.

Localización: 1A01 h.

HOOK CODE 21 h. - MOTORES DE LOS MICRODRIVES

Esta subrutina permite activar o desactivar los motores de los Microdrives. Si se selecciona un número de unidad entre 1 y 8 se activa el motor de la unidad de Microdrive indicada. En caso de seleccionar la unidad Ø se desactivan los motores de todas las unidades de Microdrive. Devolverá el mensaje de error «Microdrive not present» si la operación no puede ser llevada a cabo. Su modo de uso es:

RST 8
DEF B 21 h.

Necesita: En el registro A un número entre 1 y 8 para activar los motores de estas unidades o Ø para desactivar los motores de todas las unidades.

Devuelve: Las interrupciones desactivadas si a > Ø, o activadas si A = Ø.

Usa los registros: AF, BC, DE, HL.

Localización: 17F7 h.

HOOK CODE 22 h. - ABRE UN VIA (STREAM)

Esta subrutina permite crear un archivo en el Microdrive, devolviendo la dirección del comienzo del área del archivo en el registro IX.

Para usar esta subrutina debe asegurarse primero de que las variables del sistema existen. (Puede usar el «hook code» 31 h.).

Luego debe guardar el contenido del registro H'L' y asegurarse de colocar el nombre del archivo en el lugar correspondiente (vea el capítulo en que se habla de las variables del sistema); el número de la unidad de Microdrive en la variable del sistema DSTR 1; y la longitud del nombre del archivo en la variable del sistema NSTR 1.

Ahora ya puede usar este «hook code», y después de ello devolver el valor que tenía al registro H'L'. '

Si el nombre usado corresponde a un archivo creado anteriormente,

se interpreta como archivo de lectura y se carga el primer sector. En caso contrario se interpreta que se desea crear un nuevo archivo de escritura.

Cabe destacar que esta subrutina no para el motor del Microdrive ni testea si el cartucho que hay en éste tiene la pestaña de protección quitada. Para poder testear esto use:

```
IN A, (EF h.)
AND 1
```

Lo cual dará cero si está protegido.

El modo de uso de esta subrutina es:

```
RST 8
DEF B 20 h.
```

Necesita: Las interrupciones activadas.

Devuelve: En el registro IX la variable del sistema CURCHL (23633). En el registro HL el desplazamiento de la vía (stream) (IX-CHANS + 1). Las interrupciones activadas.

Usa los registros: AF, BC, DE, HL, B'C', D'E', H'L', IX.

Localización: 1B29 h.

HOOK CODE 23 h. - CERRAR UN ARCHIVO

Esta subrutina cierra un archivo previamente abierto. Si era un archivo de escritura, los datos contenidos en el buffer son salvados en el Microdrive y el motor de éste se para. Si el archivo era de lectura, los datos del buffer se pierden y la vía (stream) queda cerrada. Su modo de uso es:

```
RST 8
DEF B 23 h.
```

Necesita: En el registro IX el principio del área del canal del Microdrive.

Devuelve: Las interrupciones activadas.

Usa los registros: AF, BC, DE, HL.

Localización: 12A9 h.

HOOK CODE 24 h. - BORRADO EN LOS MICRODRIVES

Esta subrutina borra del Microdrive un archivo, programa o conjunto de bytes. Antes de usarla las variables del sistema D__STR1, T__STR1 y N__STR1 deben contener el número de unidad del Microdrive y el nombre del programa o archivo a borrar. Cabe destacar que esta subrutina sólo borra una vez lo especificado. Así, si se hicieron varias copias de lo que se pretende borrar, sólo será borrado una vez cada ocasión que se ejecute esta subrutina. Su modo de uso es:

RST 8
DEF B 24 h.

Necesita: NADA.

Devuelve: Las interrupciones activadas.

Usa los registros: AF, BC, DE, HL, B'C', D'E', H'L', IX.

Localización: 1D6E h.

HOOK CODE 25 h. - LECTURA SECUENCIAL

Esta subrutina se encarga de leer el siguiente sector de los que componen un archivo. Las variables del sistema CHDRIV y CHNAME deben contener el número de unidad de Microdrive y el nombre del archivo respectivamente antes de usar esta subrutina. El registro IX debe contener la dirección del inicio del área de canales del Microdrive y la variable del sistema CHREG, el número del sector que hay cargado actualmente en memoria y se incrementa al usar esta subrutina. Su modo de uso es:

RST 8
DEF B 25 h.

Necesita: En el registro IX el principio del área del canal del Microdrive.

Devuelve: Las interrupciones desactivadas. El motor del Microdrive en marcha.

Usa los registros: AF, BC, DE, HL.

Localización: 1A09 h.

HOOK CODE 26 h. - GRABACION DE UN REGISTRO

Esta subrutina graba en el cartucho del Microdrive un registro. Como es habitual, el registro IX debe contener la dirección de comienzo del área de los canales del Microdrive, que contiene los datos para el registro. Este registro es creado en el siguiente sector libre en el cartucho del Microdrive. Su modo de uso es:

RST 8
DEF B 26 h.

Necesita: En el registro IX la dirección de comienzo de los datos de los canales del Microdrive.

Devuelve: Las interrupciones activadas. El motor del Microdrive pa rado.

Usa los registros: AF, BC, DE, HL.

Localización: 11FF h.

HOOK CODE 27 h. - LECTURA ALEATORIA DE UN SECTOR DE UN ARCHIVO

Esta subrutina es similar a la del «hook code» 25 h. pero con la particularidad de que el registro que se carga es el primero que se encuentra al realizar la lectura del cartucho en el Microdrive (de un archivo determinado por las variables del sistema CHNAME y CHREG. Su modo de uso es:

RST 8
DEF B 27 h.

Necesita: En el registro IX el principio del área del canal del Microdrive.

Devuelve: Las interrupciones desactivadas. El motor del Microdrive en marcha.

Usa los registros: AF, BC, DE, HL.

Localización: 1A17 h.

HOOK CODE 28 h. - LECTURA DE UN SECTOR

Esta subrutina realiza la lectura del sector especificado por la variable del sistema CHREG. Si el sector se encuentra, se carga en memoria. Si se trata de un archivo, el flag de acarreo (carry flag) se desactiva y la subrutina retorna al lugar de llamada. Si no se trata de un archivo, el flag de acarreo se activa y la subrutina retorna. Su modo de uso es:

RST 8
DEF B 28 h.

Necesita: En el registro IX la dirección de comienzo de los datos de los canales del Microdrive.

Devuelve: Las interrupciones desactivadas. El motor del Microdrive en marcha.

Usa los registros: AF, BC, DE, HL.

Localización: 1A4B h.

HOOK CODE 29 h. - LECTURA DEL SIGUIENTE SECTOR.

Esta subrutina carga en memoria el sector siguiente al leído anteriormente. Esta operación se realiza del mismo modo que la del «hook code» anterior (28 h.). Su modo de uso es:

RST 8
DEF B 29 h.

Necesita: En el registro IX el principio del área del canal del Microdrive.

Devuelve: Las interrupciones desactivadas. El motor del Microdrive en marcha.

Usa los registros: AF, BC, DE, HL.

Localización: 1A86 h.

HOOK CODE 2A h. - ESCRITURA DE UN SECTOR

Esta subrutina realiza la acción contraria al «hook code» 28 h., es decir, realiza la grabación de un sector de acuerdo con los datos que existan en memoria. Por supuesto, las variables del sistema deben estar completadas. La grabación se realiza en el registro indicado por la variable del sistema CHREG. Su modo de uso es:

RST 8
DEF B 2A h.

Necesita: En el registro IX la dirección de comienzo de los datos de los canales del Microdrive.

Devuelve: Las interrupciones desactivadas. El motor del Microdrive en marcha.

Usa los registros: AF, BC, DE, HL.

Localización: 1A91 h.

HOOK CODE 2B h. - ABRE UNA VIA (STREAM)

Esta subrutina es similar a la del «hook code» 22 h. pero, al parecer, fue olvidada por los técnicos de **Sinclair**, ya que no funciona correctamente y por lo tanto no puede ser usada.

HOOK CODE 2C h. - BORRADO DE UN BUFFER.

Esta subrutina elimina el buffer de 595 bytes del área de los canales del Microdrive que señala la dirección indicada en el registro IX. Además se cierran todas las vías (streams). El Mapa del Microdrive (32 bytes) es borrado también si no es usado por otro canal. Así pues lo que realiza esta subrutina es bajar en 627 bytes el registro IX para que indique que no existe buffer de datos. Su modo de uso es:

RST 8
DEF B 2C h.

Necesita: En el registro IX la dirección de comienzo de los datos de los canales del Microdrive.

Devuelve: NADA.

Usa los registros: AF, BC, DE, HL.

Localización: 10C4 h.

HOOK CODE 2D h. - ABRE UNA VIA HACIA LA RED LOCAL

Esta subrutina abre una vía (stream) dirigida a la Red Local de Comunicaciones. En la variable del sistema D_STR1 debe estar el número de la estación de destino y en la variable del sistema NTSTAT, el número de estación propio. Su modo de uso es:

RST 8
DEF B 2D h.

Necesita: NADA.

Devuelve: En el registro IX la dirección del comienzo del área CHANS de la Red Local.

Usa los registros: AF, BC, DE, HL.

Localización: 0EA9 h.

HOOK CODE 2E h. - CIERRA UNA VIA (STREAM) HACIA LA RED LOCAL

Esta subrutina cierra una vía (stream) dirigida hacia la Red Local. Al usarla, si la vía (stream) estaba abierta para enviar datos, los que todavía no hubieran sido enviados, se marcan con una señal de fin de archivo y se envían a la Red Local. Si la vía (stream) era de recepción de datos, el área de datos y las variables de la vía (stream) se borran. Su modo de uso es:

RST 8
DEF B 2E h.

Necesita: En el registro IX la dirección de comienzo de los datos de los canales de la Red Local.

Devuelve: Las interrupciones activadas.

Usa los registros: AF, BC, DE, HL, IX.

Localización: 1A24 h.

HOOK CODE 2F h. - RECEPCION DE BYTES DE LA RED LOCAL.

Esta subrutina permite recibir de la Red Local un paquete de bytes. Las variables del sistema NCIRIS, NCSELF y NCNUMB deben contener los valores correspondientes para que la operación se realice correctamente. Su modo de uso es:

RST 8
DEF B 2F h.

Necesita: En el registro IX la dirección del comienzo de los datos de los canales de la Red Local.

Devuelve: Las interrupciones activadas.

Usa los registros: AF, BC, DE, HL.

Localización: 1A31 h.

HOOK CODE 30 h. - ENVIO DE BYTES HACIA LA RED LOCAL

Esta subrutina envía un paquete de bytes hacia la Red Local. Antes de usarla las variables del sistema NCOBL y NCNUMB, el contenido del buffer y la cabecera (excluyendo los sumatorios de comprobación) deben estar correctamente completados. El registro A debe contener un Ø para realizar un envío normal, o un 1 si se trata del final de un archivo (END OF FILE). Su modo de uso es:

RST 8
DEF B 30 h.

Necesita: En el registro IX la dirección de comienzo de los datos de los canales de la Red Local. En el registro A un Ø para datos o un 1 para fin de archivo.

Devuelve: Las interrupciones activadas. En el registro A el número de estación de destino.

Usa los registros: AF, BC, DE, HL.

Localización: ODB2 h.

HOOK CODE 31 h. - CREA LAS VARIABLES DEL SISTEMA

Si las variables del sistema que se añaden con el Interface 1 no han sido creadas, al usar esta subrutina son creadas. De hecho las crea la misma llamada ya que en la localización 19A8 h. sólo hay una instrucción RET. Su modo de uso es:

RST 8
DEF B 31 h.

Necesita: NADA.

Devuelve: NADA.

Usa los registros: AF, BC, DE, HL.

Localización: 19A8 h.

HOOK CODE 32 h. - LLAMADA A LA ROM «OCULTA».

Esta subrutina está reservada por **Sinclair** para futuras expansiones, pero realmente es uno de los mejores sistemas de entrar en la ROM «oculta», aunque sólo el valor del registro A puede ser pasado a la rutina. Al usarla, puede indicarse en la variable del sistema HD 11 la subrutina («hook code») a la cual se desea saltar. Para volver a activar la ROM del **Spectrum** debe efectuar una instrucción CALL 0700 h. Su modo de uso es:

RST 8
DEF B 32 h.

Necesita: En la variable del sistema HD_11 la dirección de la subrutina a la cual se desea acceder.

Devuelve: NADA.

Usa los registros: Depende de la subrutina a la que se salte.

Localización: 19A4 h.

RESUMEN DE LOS «HOOK CODES»

CODIGO	LOCALIZACION	FUNCION
1B h.	19D9 h.	Espera a que se pulse una tecla.
1C h.	19EC h.	Imprime un carácter en pantalla.
1D h.	0B81 h.	Acepta un carácter del RS-232.
1E h.	0C51 h.	Envía un carácter por el RS-232.
1F h.	19FC h.	Imprime un carácter en la ZX-Printer.
20 h.	1A01 h.	Testea el teclado.
21 h.	17F7 h.	Manejo de los motores del Microdrive.
22 h.	1B29 h.	Abre una vía (stream) al Microdrive.
23 h.	12A9 h.	Cierra una vía (stream) del Microdrive.
24 h.	1D6E h.	Borra un archivo del Microdrive.
25 h.	1A09 h.	Lee el siguiente sector de un archivo.
26 h.	11FF h.	Graba un registro en el Microdrive.
27 h.	1A17 h.	Lee aleatoriamente un sector del Microdrive.
28 h.	1A4B h.	Lee un sector determinado del Microdrive.
29 h.	1A86 h.	Lectura del siguiente sector del Microdrive.
2A h.	1A91 h.	Grabación de un sector determinado en el Microdrive.
2B h.	1B29 h.	Igual a la 22 h. pero falla.
2C h.	10C4 h.	Borrado de un buffer.
2D h.	0EA9 h.	Abre una vía (stream) hacia la Red Local.
2E h.	1A24 h.	Cierra una vía (stream) de la Red Local.
2F h.	1A31 h.	Lee un paquete de bytes de la Red Local.
30 h.	ODB2 h.	Envía un paquete de bytes hacia la Red Local.
31 h.	19A8 h.	Crea las nuevas variables del sistema.
32 h.	19A4 h.	Llama a la ROM «oculta».

CONSIDERACIONES DEL USO DEL INTERFACE 1 DESDE CODIGO MAQUINA

No olvide tener en cuenta las siguientes consideraciones cuando use el Interface 1 desde código máquina.

—Use la función USR precedida sólo de los comandos RANDOMIZE o LET, puesto que de usar otros comandos y ocurrir un error, el sistema se «rompería».

—Cuide el estado de las interrupciones en las entradas y salidas de subrutinas de la ROM «oculta». Recuerde que las interrupciones deben estar activadas siempre al volver al BASIC.

—Preserve siempre el contenido del registro H'L' cuando use las subrutinas del Microdrive si luego ha de volver al BASIC.

—Nunca use cartuchos con información importante cuando pruebe rutinas en código máquina referentes al Microdrive. Fácilmente puede borrar el contenido de estos cartuchos con un mínimo error, ya que desde código máquina no se testean las protecciones.

—Asegúrese siempre de que las nuevas variables del sistema han sido creadas cuando sea necesario.

—Por supuesto, no use rutinas en código máquina colocadas en líneas REM, cuando use el Interface 1. La creación de las zonas de trabajo y variables del Interface 1 desplazan el programa BASIC dentro de la memoria, con lo que el código máquina no queda instalado en el mismo lugar.

MENSAJES DE ERROR

Mensajes de error

El Interface 1 incorpora nuevos mensajes de error a los indicados en el Apendice B del manual de **Sinclair**. Estos nuevos mensajes, como es lógico, se refieren a las nuevas instrucciones. Son los siguientes:

CODE ERROR
Ha intentado cargar un bloque de bytes más largo de lo que usted especificó en el comando **LOAD**.

DRIVE «WRITE» PROTECTED
Ha intentado escribir en un cartucho que estaba protegido (por haber quitado la lengüeta de protección).

FILE NOT FOUND
Ha intentado cargar un archivo o programa que no existe o parte de él no puede ser encontrado. Puede ser por no haber cerrado un archivo o porque haya sido dañada al desconectar la alimentación con el cartucho en el Microdrive.

INVALID DEVICE EXPRESION
Ha usado un especificador de canal distinto de **s, p, k, m, n, t** o **b**. También se produce cuando haya colocado un punto y coma en vez de una coma usando los canales **s, p** o **k**.

INVALID DRIVE NUMBER
Ha especificado un número de Microdrive fuera del rango de 1 a 8.

INVALID NAME
Ha intentado salvar un programa o crear un fichero cuyo nombre es una cadena vacía o una cadena con más de diez caracteres.

INVALID STATION NUMBER
Ha intentado especificar un número de estación en la red local fuera del rango de Ø a 64, o fuera del rango de 1 a 64 si le ocurre al usar el comando **FORMAT**.

INVALID STREAM NUMBER

Ha intentado especificar una vía (stream) con un número fuera del rango de Ø a 15.

MERGE ERROR

Ha intentado usar el comando **MERGE** con una matriz o con un bloque de bytes. También ocurre al intentar utilizarlo con un programa salvado mediante **SAVE... LINE...**

MICRODRIVE FULL

Ha intentado escribir en un cartucho donde no había espacio libre. Si se trata de un programa puede volver a salvarlo usando un cartucho vacío. Si se trataba de un archivo de escritura deberá borrarlo por completo del cartucho al no haber espacio para colocar el sector con la marca de fin de archivo.

MICRODRIVE NOT PRESENT

Ha intentado usar un Microdrive que no está conectado al ordenador o un Microdrive que no contiene ningún cartucho.

MISSING BAUD RATE

No ha especificado la velocidad en baudios.

MISSING DRIVE NUMBER

No ha especificado el número de Microdrive.

MISSING NAME

Ha olvidado indicar el nombre del programa o archivo.

MISSING STATION NUMBER

Ha olvidado especificar el número de estación de la red local.

PROGRAM FINISHED

Ha intentado ejecutar un programa a partir de una línea mayor que la última o ha usado el comando **RUN** sin programa en memoria.

READING A «WRITE» FILE

Ha intentado leer datos de un archivo no existente o abierto para escritura.

STREAM ALREADY OPEN

Ha intentado abrir una vía (stream) para un nuevo canal que estaba

abierta. Sólo puede abrir una vía (stream) si previamente la ha cerrado.

VERIFICATION HAS FAILED

Lo que ha intentado verificar no corresponde con lo que hay en memoria.

WRITING TO A «READ» FILE

Ha intentado escribir en un archivo que ya existía. Primero debe borrar el archivo existente o crear otro nuevo.

WRONG FILE TYPE

Ha intentado usar los comandos **SAVE** o **MOVE** con un archivo no abierto. También ocurre si intenta usar el comando **MERGE** con un archivo de escritura y cuando intenta usar el comando **LOAD** con bytes o matrices como si se trataran de programas.

HOOK CODE ERROR

Este código de error no está especificado en el manual de **Sinclair** y su explicación es un tanto complicada. Aparece cuando el sistema intenta acceder a la **ROM** del Interface 1 por una dirección por la que no debe ser accedida. Es un mensaje de error que aparece en muy pocas ocasiones.

RESUMEN

Resumen de los Comandos

Resumen de los nuevos comandos que se incorporan al **Spectrum** con el Interface 1 y el Microdrive.

En este resumen:

- **n** Indica el número de Microdrive o de la estación de la red local.
- **x** Indica un número de vía (stream).
- **v** Indica una variable de cualquier tipo.
- **canal** Indica una expresión dirigida a un canal **(;«m»;1;...)**.

CAT n

Presenta en pantalla una lista en orden alfabético de hasta 50 programas o archivos que contiene el cartucho insertado en el Microdrive junto con el nombre de éste y los Kbytes que quedan libres en él.

CAT # x; n

Envía el catálogo del Microdrive n hacia la vía (stream) x.

CLOSE # x

Cierra la viá (stream) x. Si hay datos en el buffer de la vía (stream) los envía antes de cerrarlo.

ERASE «m»;n;«nom»

Borra el programa o archivo «nom» del cartucho insertado en el Microdrive n.

FORMAT «m»;n;«nom»

Formatea un cartucho insertado en el Microdrive n, con el nombre «nom».

FORMAT «n»;n
Coloca el número de estación propio de la red local a n.

FORMAT «t»; n
Coloca la velocidad en baudios **(n)** para usar en el **Interface RS-232** el canal de texto **(«t»)**.

FORMAT «b»; n
Igual que el anterior pero para usar el canal de binario **(«b»)**. Ambas velocidades en baudios deben ser iguales.

INKEY$ # X
Toma un carácter de la vía (stream) como una cadena de un sólo carácter. Si no encuentra ningún carácter en el momento de realizar la exploración de la vía (stream) asigna la cadena vacía.

INPUT # x; v
Toma información de la vía (stream) para la variable **«v»** hasta encontrar un retorno de carro. La vía (stream) debe haber sido abierta previamente para entrada de datos. Si coloca un texto o usa la coma **(,)** como separadora en el comando **INPUT #**, éste será enviado hacia la vía (stream). Puede usarse el comando **LINE** como se hacía habitualmente.

LOAD* canal
Cargar un programa, matriz de datos o un bloque de bytes del canal especificado. Sólo pueden usarse los canales **«m»**, **«n»**, y **«b»**. Al ejecutar este comando se borrará el programa que haya en memoria para cargar éste.

MERGE* canal
Tiene el mismo efecto que **LOAD *** pero sin borrar el programa que hay en la memoria, al igual que con el cassete.

MOVE fuente TO destino
Traslada un archivo de **«fuente»** a **«destino»**. Fuente y destino son vías (stream) o canales. Acaba su ejecución cuando encuentra la marca de fin de archivo en el **«fuente»**.

OPEN # vía, canal
Asigna la vía (stream) al canal especificado.

PRINT # vía...
Envía los datos **(...)** hacia la vía (stream) indicada. La via(stream) debe haber sido abierta **(OPEN)** previamente para salida de datos.

SAVE* canal
Salva un programa, matriz de datos o un bloque de bytes en el canal especificado. Sólo pueden usarse los canales **«m»**, **«n»** y **«b»**.

VERIFY* canal
Compara el programa, matriz o bloque de bytes, que deben entrar al igual que con **LOAD**, con lo que hay en la memoria del ordenador.

Las teclas de los nuevos comandos

A modo de orientación y hasta que coja un poco de práctica le incluyo un listado de las teclas que ha de pulsar para conseguir los nuevos comandos.

COMANDO	*MODO*	*TECLA*
CAT	E	SYMBOL SHIFT - 9
CLOSE #	E	SYMBOL SHIFT - 5
ERASE	E	SYMBOL SHIFT - 7
FORMAT	E	SYMBOL SHIFT - Ø
MOVE	E	SYMBOL SHIFT - 6
OPEN #	E	SYMBOL SHIFT - 4

El resto de comandos son compuestos de un comando que ya se usaba anteriormente y un signo por lo que requieren pulsar dos o más teclas.

Video-juegos

LA MINA (16K/48K)

Debes recoger las 40 bolsas escondidas en la mina y meterlas en tu carretilla. Puedes subirte en las vagonetas y ascensores, pero cuidado con los atropellos, caídas y los policías que te persiguen. 3 pantallas. Extraordinarios gráficos animados. C.M. **1.600,—**

WRECKAGE (16K/48K)

Increíblies gráficos en 3 dimensiones. Debes eliminar a las naves enemigas antes de que destruyan tu ciudad con sus bombas. Dificultad progresiva hasta lo imposible. 16 pantallas. C.M., J.K. **1.600,—**

TUNELES MARCIANOS (48K)

Recoge las siete llaves que abren el cofre del último túnel, en las docenas de túneles que debes atravesar para llegar a él. En cada uno de ellos, debes destruir a todos los guardianes para que se abran las puertas que te permiten pasar al siguiente. En alguno encontrarás vidas extras que tratarán de escaparse. Interminable sucesión de distintas criaturas. Extraordinarios gráficos animados. C.M., J.K. **1.700,—**

AUTOSTOPISTA GALACTICO (48K)

Te hallas en medio del espacio y debes recoger botellas de oxígeno para subsistir y dinero para pagar las naves-taxi. Encuentra un planeta y sitúate en él para hacer auto-stop. Hay naves-600, naves-ROLLS, naves-policía, naves-basurero, naves-taxi y naves-dudosas que te pueden robar el dinero. De este modo debes llegar al planeta cental de la inmensa galaxia ayudado de tu mapa galáctico. Extraordinarios gráficos. C.M., J.K. **1.700,—**

ROBOT FACTORY (48K)

Dirige la cadena de montaje de robots para fabricar el mayor número posible. Llena los depósitos antes de que se vacíen, ajusta la velocidad de la cadena, pero cuidado con los robots defectuosos que te atacan y las caídas. C.M., J.K. **1.700,—**

ESCALADOR LOCO (48K)

Sube por las ventanas del edificio. Cuidado con pillarte los dedos en las que se cierran, y con los tiestos, paquetes y bates de béisbol que te lanzan desde arriba. Hay diferentes zonas de ventanas donde la dificultad es mayor. C.M., J.K. 1.700,—

MANIC MINER (48K)

Número Uno de ventas en Inglaterra. Hay que pasar las 20 cavernas de la mina y recoger las llaves ara salir. Cada caverna es un juego distinto. Increíblies gráficos animados y sonido continuo. C.M., J.K. **1.700,—**

JET SET WILLY (48K)

Willy, el minero de Manic Miner, es ahora rico y vive en una lujosa mansión de 60 habitaciones. Una noche, después de una fiesta, debe recoger los vasos que han quedado dispersados por las habitaciones antes de que su ama de llaves le deje ir a la cama. Cada habitación es un juego distinto. Increíbles gráficos animados y sonido continuo. Número Uno de ventas en Inglaterra. C.M., J.K. **1.700,—**

Gestión

CONTABILIDAD PERSONAL (48K)

La manera idónea de llevar una pequeña contabilidad doméstica o del negocio. Presenta balances en 64 caracteres por línea. Salida por cualquier tipo de impresora en papel normal o ZX. **2.500,—**

CONTROL DE STOCKS 64 COLUMNAS (48K)

Todo el desarrollo del programa se maneja desde un menú en pantalla de muy fácil manejo. Presentación en pantalla de 64 caracteres por línea. Incluye control de mínimos y facturación. Salida por todo tipo de impresoras en papel normal o ZX. **3.000,—**

CONTEXT: PROCESADOR TEXTOS 64 COLUMNAS (48K)

Una aplicación casi indispensable del ZX-SPECTRUM. Realizado íntegramente en código máquina, este programa tiene la mayoría de las prestaciones de los procesadores de textos de los grandes ordenadores: justificación, caja, márgenes, inserción, copia de párrafos, edición. Además, lleva los caracteres castellanos y acentuados. Presentación en pantalla de 64 caracteres por línea y salida por cualquier tipo de impresora de papel normal o ZX. **3.000,—**

SIMPLEX: PROGRAMACION LINEAL (16K/48K)

Resolución del problema de maximización en programación lineal por el método Simplex. **2.500,—**

ASTROLOGIA (48K)

Realiza el cálculo de las posiciones planetarias (longitudes eclípticas) y de las «casas» astrológicas (por el método de Regiomontanus), representando luego todos estos datos sobre el fondo de la bóveda celeste, lo que en astrología se denomina «carta astral».

EMISION/RECEPCION MORSE (48K)

Con este programa, el ZX-SPECTRUM puede ser conectado a la salida de altavoz de un receptor MORSE y decodificarlo en pantalla. Asimismo, permite la operación a la inversa. Dispone de varias posibilidades para el archivo de textos fijos y emisión directa. La velocidad de recepción puede variar hasta un 40% sin errores en la decodificación. **2.000,—**

COL64C + LISTADOR BASIC ESPAÑOL (48K)

Convierte al ZX-SPECTRUM en un ordenador de 64 caracteres por línea en pantalla. Funciona con LIST, PRINT, AT y TAB normalmente, pero con 64 columnas en lugar de 32, y además tiene la posibilidad de listar los programas en BASIC español.

Educativos

EL CONSTRUCTOR (48K)

El popular juego de construcciones con figuras geométricas ahora con número ilimitado de piezas, goma de borrar, modelos en pantalla, etc. C.M., J.K. **1.700,—**

MAS DE 100 TÍTULOS. NOVEDADES CADA MES. DISPONIBLES EN TIENDAS ESPECIALIZADAS Y DIRECTAMENTE O POR CORREO EN ***VENTAMATIC.***

club NACIONAL DE USUARIOS DE LOS ZX

* Boletín informativo exclusivo para los socios.
* Servicio telefónico de consultas.
* Fabulosas ofertas especiales.
* Descuentos del 10% y plaza preferente en los cursos de informática de VENTAMATIC en Barcelona.
* Grandes descuentos para los socios de hasta el 35% en sus compras de ordenadores, periféricos, accesosorios y programas (SINCLAIR, COMMODORE, ORIC, etc.) a VENTAMATIC.
* Sorteos de programas en cada boletín.
* Posibilidad de comercializar los programas realizados por los socios.
* Regalo a elegir al formalizar la inscripción.



INFORMACION E INSCRIPCIONES

VENTAMATIC - Dpto. Venta por Correo - Avda. de Rhode, n.° 253 ROSAS (Gerona), o personalmente en VENTAMATIC - Exposición, Venta y Cursos - c/ Córcega, n.° 89, entlo. - 08029 BARCELONA.

TITULOS DE PROXIMA APARICION EN LA COLECCION VENTAMATIC

— LOS SUPERJUEGOS DEL ZX-SPECTRUM 16K/48K

AUTOR: ENRIQUE ENFEDAQUE

— JUEGOS, PROGRAMAS Y RUTINAS PARA EL ZX-SPECTRUM

AUTOR: LUIS AMADO

OTROS TITULOS YA PUBLICADOS

— GUIA PRACTICA DE BASIC DEL ZX81 Y DEL SPECTRUM

AUTOR: RAMON ROVIRA. 148 Págs. ISBN 84-247-0190-9. 1.200,— Ptas.

— PROGRAMACION EN CODIGO MAQUINA PARA EL ZX81 Y PARA EL SPECTRUM

AUTOR: JOAN SALES. 158 Págs. ISBN 84-247-0193-3. 1.200,— Ptas.

DISPONIBLES EN LIBRERÍAS, TIENDAS ESPECIALIZADAS, ETC. O DIRECTAMENTE Y POR CORREO EN **VENTAMATIC.**

Con este libro, el usuario del ZX-SPECTRUM que disponga también del ZX-INTERFACE 1 y del ZX-MICRODRIVE, podrá aprender a manejarlo utilizando al máximo todas sus fantásticas posibilidades.

A través de una estructuración del libro perfectamente planteada, el usuario del ZX-SPECTRUM y los ZX-MICRODRIVES irá avanzando progresivamente, mediante numerosos ejemplos prácticos, en el conocimiento de la programación y utilización de este nuevo sistema de almacenamiento de datos en masa que es el ZX-MICRODRIVE, cuya velocidad y fiabilidad es inconmensurablemente superior a la de un cassette normal. Además, aprenderá a desarrollar y usar nuevos sistemas informáticos de comunicación con periféricos de todo tipo como impresoras, plotters, modems, etc. e incluso con otros ordenadores, incluyendo otros ZX-SPECTRUM, a través de los INTERFACES SERIE RS232 y de RED LOCAL incorporados en el ZX INTERFACE 1.

Y todo ello, incluso desde CODIGO MAQUINA.

Josep-Oriol Tomas i Huynh-Quan-Chiêu

Colección Ventamatic